НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК _____004.04_____

«До захисту допущено»
Завідувач кафедри
_____ І.Р. Пархомей
(підпис)
“___”_____2019 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 121 «Інженерія програмного забезпечення»

на тему: Система детектування Deep Fake відеозаписів на основі нейронної мережі

Виконав: студент другого курсу, групи _ІТ-84мп___
(шифр групи)

Барабаш Андрій Олегович _____ _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2019 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ І.Р. Пархомей
(підпис)

«___»_____2019 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Студенту Барабашу Андрію
(прізвище, ім'я, по батькові)

1. Тема дисертації «Система детектування DeepFake відеозаписів на основі нейронної мережі» _____
науковий керівник дисертації Корнага Я. І., к.т.н., доцент _____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «___»_____ 2019 р. №_____

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження – підроблені DeepFake відеозаписи.

4. Предмет дослідження – розпізнавання DeepFake відеозаписів.

5. Перелік завдань, які потрібно розробити – аналіз проблеми та існуючих рішень; аналіз і реалізація нейронної мережі; аналіз і розробка програмного забезпечення; дослідження ефективності розробленого програмного забезпечення; дослідження ефективності розробленої нейронної мережі; маркетинговий аналіз стартап-проекту.

6. Орієнтовний перелік ілюстративного матеріалу – шість плакатів

7. Орієнтовний перелік публікацій – дві публікації

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області	13.09.2019 р.	
2	Постановка задачі	15.09.2019 р.	
3	Аналіз інформаційного забезпечення	20.09.2019 р.	
5	Аналіз алгоритмічного забезпечення	25.09.2019 р.	
6	Розробка нейронної мережі	15.10.2019 р.	
7	Розробка програмного забезпечення	01.11.2019 р.	
8	Маркетинговий аналіз стартап-проекту	10.11.2019 р.	
9	Висновки	15.11.2019 р.	

Студент _____ А. О. Барабаш
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ Я. І. Корнага
(підпис) (ініціали, прізвище)

АНОТАЦІЯ

У роботі розглянуто проблему небезпеки громадян через шахрайство за допомогою підробки відеозаписів нейронними мережами. Показано основні існуючі шляхи вирішення проблеми використаних в аналогічних продуктах, їх переваги та недоліки.

Розроблено нейронну мережу, що забезпечує детектування штучно підроблених відеозаписів з найкращою точністю на рівні 94 %, а також систему для кінцевого користувача, яка відкриває доступ до використання нейронної мережі без розкриття деталей її реалізації. Данна система може бути використана для детектування Deep Fakes відеозаписів з метою захисту громадян від факту шахрайства такими відео.

Ключові слова: згорткова нейронна мережа, глибокі підробки, розпізнавання обличчя, машинне навчання.

Розмір пояснювальної записки – 100 аркушів, містить 15 ілюстрацій, 23 таблиці, 2 додатка.

ABSTRACT

The thesis examines the problem of the danger of citizens through fraud by means of falsification of video with neural networks, shows the main existing ways of solving the problem used in similar products, their advantages and disadvantages.

Developed a neural network that provides detection of artificially faked videos with the best accuracy of 94 %, as well as an end-user system that gives access to the use of the neural network without revealing details of its implementation. Created system could be used to detect Deep Fakes videos for protecting citizens from the fraud.

Keywords: Convolutional neural network, DeepFakes, Face Recognition, Machine Learning.

Explanatory note size - 100 pages, contains 15 illustrations, 23 tables, 2 appendices.

Пояснювальна записка
до магістерської дисертації

на тему: ***Система детектування Deep Fake відеозаписів на основі нейронної мережі***

Київ – 2019 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Deep Fake – глибока підробка, відеозапис штучно сфабрикований за допомогою нейронної мережі, в якому обличчя однією людини заміняється на обличчя іншої;

AI – artificial intelligence, штучний інтелект;

ЦО – цільова особа, людина яку зображено на відеозаписі;

ОД – одиниця дії обличчя, характерні рухи обличчя цільової особи;

ROC – receiver operating characteristic, робоча характеристика приймача. Графік кривої, що дозволяє оцінити якість бінарної класифікації;

AUC – area under ROC curve, площа під ROC-кривою. Міра вимірювання точності розробленої нейронної мережі;

AWS – Amazon Web Services, платформа хмарних обчислень від компанії Amazon;

EKS – Elastic Kubernetes Service, хмарна платформа компанії Amazon для розгортки і управління кластером Kubernetes.

ВСТУП

Створення складних підроблених відеозаписів довгий час залишалось прерогативою лише великих голлівудських кіностудій та пародійних акторів. Проте, сучасні відкриття у глибокому навчанні зробили набагато легшим процес створення складних і одночасно правдоподібних фейкових відеозаписів. З відносно великою кількістю відкритих даних, доступних кожному, та сучасними обчислювальними потужностями, звичайна людина може, наприклад, створити відеозапис на якому один із світових лідерів зізнається у незаконній діяльності, що далі призводить до конституційної кризи, або наприклад, військовий лідер робить агресивну заяву, що призводить до громадянських заворушень у зоні військової активності. Або гігант корпоративного ринку стверджує, що їх прибутки слабшають, що призводить до глобальних маніпуляцій з акціями. Ці так звані глибокі підробки, або Deep Fakes, становлять значну загрозу нашій демократії, національній безпеці та суспільству.

Deep Learning, або глибинне навчання, породило технології, які вважалися неможливими лише кілька років тому. Сучасні генеративні моделі є одним із таких прикладів, вони здатні синтезувати гіперреалістичні образи, мовлення, музику та навіть відео. Ці моделі знайшли застосування в найрізноманітніших програмах, включаючи підвищення доступності для людей з вадами за допомогою текстового мовлення та сприяння створенню навчальних даних для медичних зображень.

Але популяризація нейронних мереж несе не тільки користь людству, а може нанести і шкоду. Яскравим прикладом цього є глибинні фейки. Термін Deep Fakes з англійської – це поєднання слів «глибинне навчання» (Deep Learning) та «фейк» (Fake). Це відеозаписи на яких за допомогою нейронних мереж змінено обличчя людини на інше обличчя. Таким чином глядач може бути введений в оману, а такі відеозаписи можуть зіпсувати репутацію людині, або й нести реальну кібер-загрозу.

Вперше Deep Fake відео з’явились в 2017 році після серії фільмів для дорослих, де замість акторок були нібито справжні зірки або героїні кінофільмів. У лютому 2018 року популярний веб-сайт Reddit забанив відповідну спільноту за поширення фейкових відео. Проте автор таких відеозаписів встиг поширити свою програму для створення таких стрічок під назвою FakeApp.

Відомі люди найчастіше стають об’єктами Deep Fakes, тому що у вільному доступі існує багато світлин та відео з їхніми обличчями. Нейронна мережа навчається на основі цих зображень, створює маску з обличчям знаменитостей, після чого вона можна замінювати будь-яке обличчя у вже відзнятому відео на згенероване. Те саме стосується й інших публічних осіб, наприклад, політиків. Втім, якщо ви публікуєте багато своїх власних зображень в інтернеті, ви також можете стати об’єктом відеопідробки.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Постановка і актуальність задачі

Deep Learning, як і будь-яка сучасна та стрімко зростаюча технологія, створила нові виклики. Так звані «Deep Fakes» або «глибинні фейки» - створені за допомогою глибинних генеративних моделей, які можуть маніпулювати відео та аудіокліпами – одні з них. З часу їх першої появи наприкінці 2017 року з’явилося багато методів генерації глибинних фейків із відкритим кодом, що призвело до збільшення кількості синтезованих медіа-роликів. Хоча багато хто, ймовірно, має намір бути жартівливим, інші можуть бути шкідливими та небезпечними для людей і суспільства.

Зловмисник може за допомогою готових програм з відкритим кодом легко замінити обличчя однієї діючої особи на іншу, причому кінцевий глядач навіть не замислиться про те, що даний відеоролик може бути підробкою. Deep Fakes можуть бути використані терористами, або злочинними групами у власних цілях. Можна наприклад, замінити промову президента на промову радикалу, що може призвести до невідомих наслідків, або ж підробити показання людини для судового рішення. Можливості глибинних фейків майже необмежені, що робить їх неймовірно небезпечною технологією.

З початку появи Deep Fakes пройшло всього два роки, проте дана технологія розвивається дуже швидкими темпами. Так, на зорі появи технології усі перероблені відео виглядали безглуздо і кумедно, можна було легко зрозуміти де підробка. Проте глибинні фейки не стояли на місці і за два роки перетворились у надзвичайно якісну технологію. На рис. 1.1 можна побачити кадр із фільму Стенлі Кубрика – Сяйво. На одній стороні глибинна копія, на іншій – оригінальний кадр із фільму. Якщо людина не дивилась дане кіно, то навряд чи зможе відрізнити підробку від оригіналу. Який кадр справжній, з Джимом Керрі чи з Джеком Ніколсоном? Кадр справа оригінальний, а зліва –

підробка. Проте людина котра не порівнює їх одна до одної ніколи б не подумала, що на лівому кадрі глибинний фейк.

Рис. 1.1. Порівняння оригіналу і Deep Fake

На допомогу непідготовленому людському оку приходить, власне, те що і створило Deep Fakes – нейронні мережі. Якщо людина, в силу власних особливостей, може навіть просто не задуматись про те, що на екрані може бути підробка, то нейронна мережа працює завжди і її дуже важко провести.

У січні 2018 року було запущено настільну програму FakeApp. Застосунок дозволяє користувачам легко створювати та обмінюватися відео, обмінюючись обличчям. Додаток використовує штучну нейронну мережу та потужність графічного процесора та три-чотири гігабайти пам'яті для створення фальшивого відео. Для отримання детальної інформації програмі потрібна велика кількість візуальних матеріалів від людини, яку потрібно вставити, щоб дізнатись, які аспекти зображення мають бути замінені, використовуючи алгоритм глибокого навчання на основі відеопослідовностей та зображень. Програмне забезпечення

використовує Framework штучного інтелекту TensorFlow від Google, який, серед іншого, вже використовувався для програми DeepDream. Знаменитості є основними цілями подібних фальшивих секс-відео, але деякі звичайні люди також постраждали. У серпні 2018 р. дослідники у Університеті Каліфорнії, Берклі опублікували статтю, в якій представлено додаток, який може замінити дитину експериментальною танцівницею, використовуючи штучний інтелект.

Інший ефект Deep Fake полягає в тому, що більше не можна розпізнати, чи є вміст підготовленим (наприклад, сатира), або справжнім. Дослідник штучного інтелекту Alex Champandard сказав, що всі повинні знати, як сьогоднішня ситуація може бути пошкоджена з використанням цієї технології, і що проблема не є технічною, а, швидше за все, вирішена шляхом довіри до інформації та журналістики. Первинним є те, що людство може потрапляти в епоху, в якому більше неможливо визначити, чи відповідає істині те, що показується у ЗМІ.

Дана проблема не оминула навіть гігантів ринку технологій. Наприклад, компанія Google сприймає нейронні мережі і їх небезпечність дуже серйозно. Минулого року вони випустили так звані AI Principles – набір правил, котрих варто притримуватись при розробці нейронних мереж задля безпеки і надійності для людства. У січні 2019 року Google анонсувала випуск величезного датасету з штучно створеними голосовими записами, щоб команди вчених і різноманітні організації змогли навчитись відрізняти Deep Fake аудіо-записи.

Отже, проблема детектування Deep Fake відеозаписів є актуальною на сьогоднішній день, а задачею даної дипломної дисертації є:

1. Побудова оптимальної моделі нейронної мережі;
2. Знаходження або створення датасету з оригінальним і Deep Fake відео;
3. Навчання нейронної мережі;
4. Створення зручної системи для кінцевого користувача;
5. Інтегрування нейронної мережі до системи.

1.2 Огляд існуючих рішень

Існує велика кількість літератури з питань зображень та відео криміналістики. Але, оскільки штучно синтезований контент є відносно новим явищем, існує недостатність криміналістичних методик конкретного виявлення глибоких підробок. Один з прикладів такої методики ґрунтується на розумному спостереженні, що люди, зображені в першому поколінні глибоких підробок або не моргали, або блимали на очікуваній частоті. Цей недолік штучних відео був пов'язаний з тим, що дані, що використовуються для синтезу обличчя, як правило, не зображували людину із заплющеними очима. Дещо передбачувано, незабаром після оприлюднення цієї криміналістичної методики, наступне покоління методів синтезу включило блимання у свої системи, так що ця методика зараз є менш ефективною. Ця ж команда також розробила методику для виявлення глибоких підробок, використовуючи відмінності в оціненій тривимірній позі голови, що обчислюється з особливостей по всьому обличчю та особливостей лише в центральній (потенційно зміненій) області обличчя. Незважаючи на те, що він ефективний при виявленні змін обличчя, цей підхід не ефективний при виявленні глибинних підробок синхронізації губ або майстра ляльок.

Інші криміналістичні методи експлуатують артефакти пікселів низького рівня, введені під час синтезу. Хоча ці методи виявляють різноманітні підробки з відносно високою точністю, вони страждають, як і інші методи, засновані на пікселях, від простих заходів протидії відмиванню, які можуть легко знищити вимірюваний артефакт (наприклад, шум присадки, рекомпресія, зміна розміру). Описано криміналістичну техніку, яка призначена для виявлення глибоких підробок людини. Наша криміналістична техніка підлаштовується для конкретних осіб і, через ризик для суспільства та демократичних виборів, орієнтуємося на світових та національних лідерів та кандидатів на високі посади. Зокрема, показано, що коли люди говорять, вони виявляють відносно чіткі

моделі руху обличчя та голови (див. приклад, а також, в яких рухи верхньої частини тіла використовувались для ідентифікації оратора). Також показано, що створення всіх трьох типів глибоких підробок має тенденцію порушити ці зразки, оскільки вирази контролюються імітатором (обміном обличчя та майстром ляльок) або рот відокремлюється від решти обличчя (синхронізація губ). Використано ці закономірності, будуючи те, що називається м'якими біометричними моделями високопоставлених осіб, і використовуємо ці моделі для розрізнення реальних та фальшивих відео. Продемонстровано ефективність такого підходу на великій кількості глибоких підробок низки американських політиків, таких як Хілларі Клінтон, Барака Обами, Берні Сандерса, Дональда Трампа та Елізабет Воррен. Цей підхід, на відміну від попередніх підходів, стійкий до відмивання, оскільки він спирається на порівняно грубі вимірювання, які не зруйнуються легко, і здатний виявити всі три форми глибоких підробок.

Розроблювана система в контексті даної магістерської дисертації є не першою системою, мета котрої відрізнити підроблений відеозапис від справжнього. На сьогоднішній день існує декілька систем, котрі спрямовані вирішити дану проблему. Розглянемо їх детальніше.

Команда ентузіастів з Інституту Інформаційних Наук при Технічному Коледжі в Університеті Південної Каліфорнії (ориг. USC Information Sciences Institute (USC ISI)) були першими в цій гонці. Вони розробили систему, котра розпізнає обличчя на відеозаписах і стежить за ним. Вона помічає нехарактерні для справжньої людини лицеві рухи, а це притаманно Deep Fake відеозаписам, і на основі цих даних приймає рішення про справжність відеоролика. Така система добре працює з неякісно зробленими глибинними фейками, як наприклад на рис. 1.2. В цьому прикладі можна побачити, що лінії обличчя і лінії голови не співпадають, вони зміщені як показано вектором. Таким чином система Каліфорнійського Інституту Інформаційних Наук має плюси, але і мінуси також. До перших можна віднести те, що ця система першою показала світові

потрібність боротися з глибинними фейками і показала, що це може робити машина, а не тільки людина.

Рис. 1.2. Неякісно заміщене обличчя Ілона Маска
на обличчя Ніколаса Кейджа

До мінусів же відноситься те, що даний алгоритм не може відрізнити якісно зроблену копію, як на рис. 1.3, а також закритість системи – розробники так і не випустили відкриту версію для усіх бажаючих.

Рис. 1.3. Підроблений президент Габону

Пізніше в гонку розпізнавання відео включились вчені із Каліфорнійського університету в Берклі. Їх система так само націлена на детектування Deep Fake відеозаписів, проте має інший алгоритм. Розробники створили програму, з нейронною мережею у своїй основі, яка вивчає відеозаписи відомих людей і запам’ятовує їх поведінку і емоції. Далі, коли треба перевірити відео на справжність, то алгоритм спочатку визначає людину на відео (не важливо підроблена ця людина, або справжня), і потім порівнює характерні риси конкретної людини на відео з даними, котра нейронна мережа вивчила раніше. На основі цього робиться рішення – дане відео глибинний фейк чи ні. Такий підхід безперечно має свої плюси, наприклад ця система дає безпеку високопосадовцям – зловмисник не зможе використати обличчя президента, наприклад, у своїх цілях. Проте ця система не позбавлена мінусів – вона не працює для більшості населення, адже націлена тільки на відомих людей, а також з розвитком Deep Fake відеозаписів цей алгоритм буде працювати гірше, адже технології заміщення обличчя не стоять на місці і розвиваються – вже є приклади таких підробок з дуже реалістичними емоціями.

1.3 Вимоги до системи

Загальні вимоги до нейронної мережі:

- Мати точність виявлення фейкових відео на рівні 90% і вище;
- Мати можливість навчатись не за допомогою датасету, а за допомогою відео, котрі хоче перевірити кінцевий користувач;
- Мати допустимі норми швидкості перевірки 1 відеозапису.
- Загальні вимоги до системи для кінцевого користувача:
- Мати можливість завантажувати власне відео для перевірки;
- Мати можливість передавати відео на перевірку за посиланням без необхідного завантаження;

- Мати можливість зареєструвати користувача, для збереження історії перевірки відеозаписів. Дану функцію користувач може вимкнути за бажанням.
- Мати простий і зручний користувацький інтерфейс;
- Мати API для інтегрування системи в інші програми.

Система має складатися з:

- Підсистеми для реєстрації користувача;
- Підсистеми для перевірки відео;
- Підсистеми для взаємозв’язку з нейронною мережею;
- Підсистеми обрізання відео для оптимальної швидкості алгоритму
- Підсистема для відображення та редагування інформації про користувача

Доступ до функціоналу системи має здійснюватися через уніфікований програмний інтерфейс, що має відповідати архітектурному стилю REST. Для відображення системи користувачу використовуватиметься веб-інтерфейс та мобільні додатки.

Висновки до розділу

Отже, розвиток нейронних мереж є настільки швидким, що з’явилися системи, котрі можуть нашкодити людству. Однією з таких технологій є Deep Fakes, або глибинні фейки – відеозаписи на котрих обличчя людини заміняється на обличчя іншої людини, причому усі емоції, вирази обличчя і рухи залишаються, а підробку майже не помітно.

Тому на сьогоднішній день є актуальною проблема відокремлення фейкових відеозаписів від оригінальних. Розроблювана система детектування Deep Fake відеозаписів за даною магістерською дисертацією націлена на вирішення даних питань. Її можна вирішити двома шляхами – алгоритмами відстеження обличчя і лицевих рухів, або нейронною мережею. Обрано другий варіант, адже дана система швидше працює, а також простіша для розробки за наявності потрібного датасету.

РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Алгоритм виділення обличчя

Розпізнавання обличь за допомогою комп'ютера було вперше запропоновано в 1966 р Woody Bledsoe і ін. У першій системі використовувався графічний планшет, на якому було необхідно вручну відзначити ключові особливості на обличчі (брови, губи і тд). Система подальшому використовувала відстані між цими точками для порівняння різних зображень. Оскільки потрібно багато ручної роботи, то оператор міг обробити близько 40 зображень в годину. У 1997 була створена система, яка безпосередньо працювала з відеоданими і могла застосовуватися на практиці. В далі зі збільшенням продуктивності обчислювальних систем з'явилися більш якісні алгоритми, але вони помітно поступалися людині на цьому завданню. З появою потужних відеокарт в 2014 році з'явилися системи, точність яких вже перевищувала точність оператора (людини). Основна перевага таких систем в тому, що вони можуть обробляти великі обсяги даних - продуктивність людини набагато нижче (потрібно кілька секунд на порівняння пари зображень).

Завдання ізоляції особистості людини у природному чи штучному середовищі та подальшої ідентифікації завжди було одним із найважливіших завдань для дослідників, які працюють у сфері машинного зору та штучного інтелекту. Однак багато досліджень, проведених у провідних наукових центрах світу протягом кількох десятиліть, не призвели до створення реальних систем комп'ютерного зору, здатних виявляти та розпізнавати людей у будь-якому середовищі. Незважаючи на схожість завдань і методів, що використовуються при розробці альтернативних систем біометричної ідентифікації, таких як ідентифікація за відбитками пальців або зображення райдужної оболонки, система ідентифікації на зображенні людини істотно поступається вищевказаним системам.

Серйозна проблема, з якою стикаються системи комп'ютерного зору, - велика мінливість візуальних зображень, пов'язана зі змінами світла, кольору, масштабу та перспективи. Крім того, у людей є звичка гуляти вулицями та в приміщенні, що призводить до значної мінливості образів тієї самої людини. Однак найскладнішим завданням комп’ютерного зору є проблема неоднозначності, яка виникає при проєктуванні тривимірних об'єктів реального світу на плоскі зображення. Колір та яскравість окремих пікселів зображення також залежить від багатьох важко передбачуваних факторів. До таких факторів належать:

- кількість та розташування джерел світла;
- колір та інтенсивність випромінювання;
- тіні або відображення від навколишніх предметів.

Завдання виявлення об’єктів на зображенні також ускладнюється величезною кількістю даних, що містяться в зображенні. Зображення може містити тисячі пікселів, кожен з яких може мати важливе значення. Повне використання інформації, що міститься на зображенні, вимагає аналізу кожного пікселя для його об'єкта чи фону з урахуванням можливої мінливості об’єктів. Такий аналіз може зажадати великих витрат на необхідну пам'ять та продуктивність комп'ютера.

Рішення цієї проблеми полягає в правильному виборі опису об'єктів, для виявлення та розпізнавання яких створена система. Опис об’єкта має враховувати найхарактерніші особливості та бути достатньо репрезентативним, щоб відрізняти цей об’єкт від інших елементів оточуючої сцени. Щоб уникнути суб’єктивності при виборі потрібного опису, можна використовувати методи автоматичного відбору відповідних характеристик об'єкта, які реалізовані в генетичних алгоритмах та в тренуванні штучних нейронних мереж. У той же час в описі об'єкта є ряд параметрів, які слід обрати досліднику, що розробляє систему виявлення та розпізнавання. Цей вибір включає:

- вибір між 2D та 3D-зображенням сцени та об'єкта. Алгоритми, що використовують 2D-представлення, як правило, простіші, ніж тривимірні алгоритми, але в той же час вони вимагають великої кількості різних описів, що відповідають представленню об'єкта в різних умовах спостереження;
- вибір між описом об’єкта в цілому або як системи, що складається з певного набору взаємопов'язаних елементів;
- вибір між системою ознак на основі геометричних або інших характеристик, що описують специфіку об'єкта.

У найзагальнішому випадку алгоритм вирішення проблеми виявлення та ідентифікації людини з зображення його обличчя складається з наступних очевидних кроків:

1. Виявлення факту присутності людини в аналізованій сцені;
2. Виділення людської фігури;
3. Головний розряд;
4. Визначення кута спостереження голови (повне обличчя, профіль);
5. Виділення обличчя;
6. Порівняння зі стандартами та ідентифікацією.

Залежно від конкретних умов структура та реалізація окремих етапів алгоритму можуть відрізнятися. У найскладнішому випадку, коли використовується система виявлення та ідентифікації людини за зображенням її обличчя у сильно мінливому середовищі, з великим потоком вхідних даних (робота на міських вулицях з великим трафіком, в метро, аеропортах, тощо), використання найбільш доступної інформації для досягнення задовільних результатів алгоритму. Алгоритм повинен вміти ефективно відрізати статичні і повільно мінливі елементи сцени, працювати в різних умовах освітлення, розпізнавати фігуру людини з різних ракурсів, відстежувати рух багатьох людей і автоматично обирати момент, відповідний для ідентифікації даної людини (наприклад , коли можна отримати фронтальне зображення людини з достатньою

роздільною здатністю). Для забезпечення таких можливостей алгоритму необхідне певне апаратне насичення системи, включаючи огляд мультикамер та аналіз сцени з можливістю виділення тривимірної структури сцени, високошвидкісний вхід відеопотоку для фільтрації елементів сцени за параметрами руху, використовуючи колір, щоб виділити елементи сцени. Крім того, потрібні камери з високою роздільною здатністю та хорошою оптикою для забезпечення максимально можливого діапазону надійної ідентифікації. У більш простих випадках, при статичній сцені та обмеженому потоці подій (подій людей) можливо використовувати більш просту структуру апаратури та алгоритм, наприклад, стереопару або одну камеру та заздалегідь підготовлену модель сцени. бути достатнім, щоб достовірно визначити, чи перебуває людина в зоні управління, підкресливши свою фігуру та особу.

Завдання визначення факту присутності людини на сцені вимагає певного рівня інтелекту від алгоритму. Це не повинна бути система, яка реагує просто на те, що змінюється сцена. Алгоритм виявлення людини не повинен видавати помилкові тривоги, коли відбувається зміна освітленості, переміщення тіней від статичних об'єктів, поява тварин у зоні управління тощо. При необхідності виникає проблема створення адекватного опису сцени. Цей опис може представляти тривимірну модель сцени, імовірнісну модель розподілу кольорів чи яскравості елементів сцени або систему знаків, що відрізняє елементи сцени від об'єктів розпізнавання (у нашому випадку - людські фігури). Відносини між елементами сцени, які вважаються фоном або елементами переднього плану, можуть змінюватися. Цю ж фігуру людини, якщо її зображення менше певного порогового значення, визначеного роздільною здатністю оптичної системи, можна віднести до фонових елементів, оскільки її аналіз малопродуктивний для виконання головного завдання - ідентифікації людини.

Вибір алгоритму, який використовується для ідентифікації людини за зображенням її обличчя, також залежить від конкретних умов його застосування. Наприклад, багатошарова нейронна мережа може легко впоратися із завданням

розпізнавання в суворо обмеженій команді. У той же час завдання виявлення конкретної людини в натовпі (з невизначеним складом) вимагає використання складних методів для зменшення помилкових тривог. Швидше за все, у цьому випадку знадобиться багаторівнева система, що містить безліч аналізаторів, що працюють в різних просторах атрибутів, з прийняттям рішення методом голосування. На початкових етапах роботи система ідентифікації повинна відрізати явно непридатних кандидатів та використовувати решту наборів кандидатів для прийняття остаточного рішення щодо ідентифікації.

На сьогоднішній день найрозповсюдженішим методом розпізнавання обличчя є метод з використанням згорткової нейронної мережі. Алгоритм складається з побудови біометричних шаблонів (дескрипторів) по зображенню і пошуку заданого шаблону в базі вже обчислених дескрипторів. Класичні архітектури нейронних мереж з повнозв'язними шарами незастосовні на практиці для аналізу зображень, оскільки кількість параметрів такої мережі експоненціально зростає з розміром вхідних даних і кількістю шарів. У таких мережах кожен нейрон не пов'язаний з усіма нейронами попереднього шару, а домножується на деякий коефіцієнт (ядро згортки) по всьому зображенню. Таким чином, кожен такий шар застосовує операцію згортки до входів зі своїми коефіцієнтами, що помітно зменшує число ваг в нейронної мережі. Це дозволяє зменшити споживання пам'яті та ефект перенавчання мережі, який проявляється в тому, що мережа добре працює на класифікації елементів навчальної вибірки і гірше на реальній.

2.2 Алгоритм відстеження та вимірювання обличчя

Для вирішення задачі використано інструмент аналізу обличчя з відкритим кодом OpenFace2 для витягування рухів обличчя та голови у відео. Ця бібліотека містить 2-D та 3-D орієнтири для обличчя, позу голови, очні погляди та одиничні дії для кожного кадру в даному відео.

Рухи м’язів обличчя можна представити у вигляді одиниць дії обличчя (ОД). Інструментарій OpenFace2 забезпечує інтенсивність та появу для 17 ОД: внутрішній підйом брови (ОД01), зовнішній підйом брови (ОД02), опущення брови (ОД04), підйом верхньої повіки (ОД05), підйом щоки (ОД06), стискання повіки (ОД07), морщення носа (ОД09), підйом верхньої губи (ОД10), піднімання куточка губи (ОД12), ямочка (ОД14), опускання куточка губи (ОД15), підйом підборіддя (ОД17), розтягування губ (ОД20), стискання губ (ОД23), розведення двох губ (ОД25), опускання щелепи (ОД26) та моргання очей (ОД45).

Розроблювана модель містить 16 ОД – ОД миготіння очей було усунене, оскільки було виявлено недостатньо відмінним для наших цілей. Ці 16 ОД доповнені такими чотирма ознаками: (1) обертання голови навколо осі x (кивання); (2) обертання голови навколо осі z (нахил); (3) 3-D горизонтальна відстань між куточками рота (potℎ); та (4) 3-D вертикальна відстань між нижньою та верхньою губою (potv). Перша пара особливостей фіксує загальний рух голови (ми не вважаємо обертання навколо осі y (поворот) через відмінності, коли йде розмова безпосередньо з людиною та з великим натовпом). Друга пара цих функцій фіксує розтягнення рота (ОД27) та смоктання губ (ОД28), які не охоплюються нашими стандартними 16 ОД.

В роботі використано кореляцію Пірсона для вимірювання лінійності між цими ознаками, щоб охарактеризувати особистий рух людини. Маючи в цілому 20 функцій обличчя / голови, обчислюємо кореляцію Пірсона між усіма 20 цими характеристиками, отримуючи

$$20C2 = (20 \times 19) / 2 = 190 \text{ пар функцій}$$

у всіх 10-секундних відеокліпах, що перекриваються. Кожен 10-секундний відеокліп зводиться до характеристичного вектору розмірністю 190 елементів, який, як описано далі, потім буде використовуватися для класифікації відео як справжнього або підробленого.

2.3 Набір даних для навчання

У роботі найбільше зосереджено увагу на відеозаписах цільових осіб (ЦО), які розмовляють у формальній обстановці, наприклад, щотижневі розмови, інтерв'ю з новинами та публічні виступи. Усі відео завантажувались вручну з YouTube, де увага ЦО в першу чергу спрямована на камеру. Для кожного завантаженого відео вручну витянуто сегменти відео, які відповідали таким вимогам: (1) сегмент становить не менше 10 сек; (2) ЦО розмовляє протягом усього сегменту; (3) у сегменті видно лише одне обличчя - ЦО; і (4) камера є відносно нерухомою протягом сегмента (дозволено повільне збільшення). Всі сегменти були збережені при 30 кадрів в секунду, використовуючи mp4-формат при відносно високій якості 20. Кожен сегмент потім був розподілений на 10-секундні кліпи, що перекриваються (кліпи витягувались, просуваючи вікно по п'яти кадрах сегменту одночасно). Перевірено розроблюваний підхід за допомогою таких наборів даних: 1) 5,6 годин відео-сегментів 1 004 унікальних людей, що дають 30, 683 10-секундних кліпів із набору даних FaceForensics; 2) комедійні імітатори для кожної ЦО; 3) глибокі підробки з підміною обличчя, з синхронізацією губ та глибокі підробки методом майстра ляльок. На рис. 2.1 показано п’ять прикладів кадрів із 10-секундного кліпу оригінального відео з Бараком Обамою, підробки методом синхронізації губ, методом комедійного імітатора, методом обміну обличчям, а також методом майстра ляльок.

Отже, створено датасет з цільовими особами (ЦО), котрий сформовано за допомогою відеороликів з порталу YouTube. Далі, ці ролики відформатовано так, щоб скласти сегменти відео по 10 секунд, які накладаються один на одного. Таким чином, створено більше інформації для навчання нейронної мережі. Другим датасетом є нещодавно відкритий датасет оригінальних і Deep Fake відеозаписів від компанії Google. Для навчання використані обидва набори даних.

Рис. 2.1 Згори донизу: оригінальний відеозапис, глибинна копія методом синхронізації губ, методом імітатора, обміну обличчям і майстром ляльок

2.4 Глибинні підробки

Використовуючи відеозаписи їхніх комедійних імітаторів як основу, створено глибокі підробки обличчя для кожної ЦО. Для обміну обличчями між кожною ЦО та їх імітатором було підготовлено генеративну змагальну мережу (GAN), засновану на архітектурі Deep Fake. Кожна GAN навчалася приблизно з 5000 зображеннями на ЦО. Потім GAN замінює обличчя імітатора обличчям ЦО, співставляючи вираз імітатора та позу голови на кожному кадрі відео. Спочатку виявлено орієнтири обличчя та рамки для обмеження обличчя за допомогою dlib (бібліотека). Центральні 82% обмежувального поля використовуються для формування обличчя ЦО. Потім сформоване обличчя вирівнюється з оригінальним обличчям за допомогою орієнтирів обличчя. Контур орієнтиру обличчя використовується для створення маски для післяобробки, яка включає

альфа-змішування та відповідність кольорів для поліпшення просторово-часової узгодженості остаточного відеозапису обличчя.

Використовуючи комедійних імітаторів Барака Обами, також створено глибокі підробки методом майстра ляльок для Обами. Фотореалістичний аватар GAN (paGAN) синтезує фотореалістичні обличчя з однієї картини. Цей процес створює відео лише плаваючої голови на статичному чорному тлі. Окрім створення таких підробок, модифіковано цей процес синтезу, видаляючи маски для обличчя під час тренувань, дозволяючи створювати відео з непошкодженим фоном. Тимчасова узгодженість цих відео була покращена за рахунок кондиціонування мережі з декількома кадрами, що дозволяло мережі вчасно бачити. Ця модифікована модель була навчена, використовуючи лише зображення Барака Обами.

Хоча обидва ці типи підробок є візуально переконливими, вони іноді містять просторово-часові глюки. Ці глюки, однак, постійно зменшуються, і є впевненість що відеоролики майбутніх версій будуть з невеликими або відсутніми помилками.

Висновки до розділу

Отже, після дослідження існуючих алгоритмів виділення обличчя із зображення, було обрано алгоритм згорткової нейронної мережі.

Для вимірювання і відстеження обличчя обрано алгоритм виділення одиниць дії обличчя. Використано кореляцію Пірсона для вимірювання залежності між цими ознаками.

В якості набору даних для навчання використані скачані відеоролики відомих особистостей з порталу YouTube. Кожне відео розділене на сегменти по 10 секунд.

Для перевірки алгоритму створено набір Deep Fake роликів за участі Барака Обами усіма існуючими методами.

РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Архітектура програмного забезпечення

Архітектура – це структура програмної чи комп’ютерної системи, яка визначає її роботу на найвищому концептуальному рівні, включаючи апаратні та програмні компоненти, властивості цих компонентів, які видно зовні, зв’язки між ними та документування системи. Документування архітектури спрощує процес взаємодії між учасниками проєкту, дозволяє фіксувати рішення, прийняті на ранніх етапах проєктування, для проєктування системи високого рівня та неодноразово використовувати елементи цього дизайну та шаблони в інших проєктах.

Будучи в момент свого розвитку дисципліною без чітких правил щодо "правильного" способу створення системи, дизайн архітектури програмного забезпечення все ще є сумішшю науки і мистецтва. Аспект "мистецтва" полягає в тому, що будь-яка комерційна система передбачає застосування або місію. Які ключові цілі системи описані за допомогою скриптів як нефункціональних вимог до системи, також відомих як атрибути якості, що визначають, як система буде вести себе. Атрибути якості системи включають в себе відмовостійкість, підтримуючи зворотну сумісність, розширюваність, надійність, ремонтопридатність, доступність, безпеку, зручність використання та інші якості. З точки зору користувача архітектури програмного забезпечення, архітектура програмного забезпечення забезпечує напрямок руху та вирішення проблем, пов'язаних зі спеціальністю кожного такого користувача, наприклад, зацікавлена особа, розробник програмного забезпечення, група підтримки програмного забезпечення, програмне забезпечення спеціаліст, фахівець із розробки програмного забезпечення, тестер та кінцеві користувачі. У цьому сенсі архітектура програмного забезпечення фактично поєднує різні точки зору на

систему. Те, що ці декілька різних точок зору можуть поєднуватися в архітектурі програмного забезпечення, є аргументом на захист необхідності та доцільності створення архітектури програмного забезпечення ще до стадії розробки програмного забезпечення.

Для успішного виконання поставленої задачі, розроблювана система має відповідати наступним вимогам:

- Доступність
- Надійність
- Зручність

Для цих цілей розроблено наступну структуру вихідного програмного забезпечення (рис.3.1).

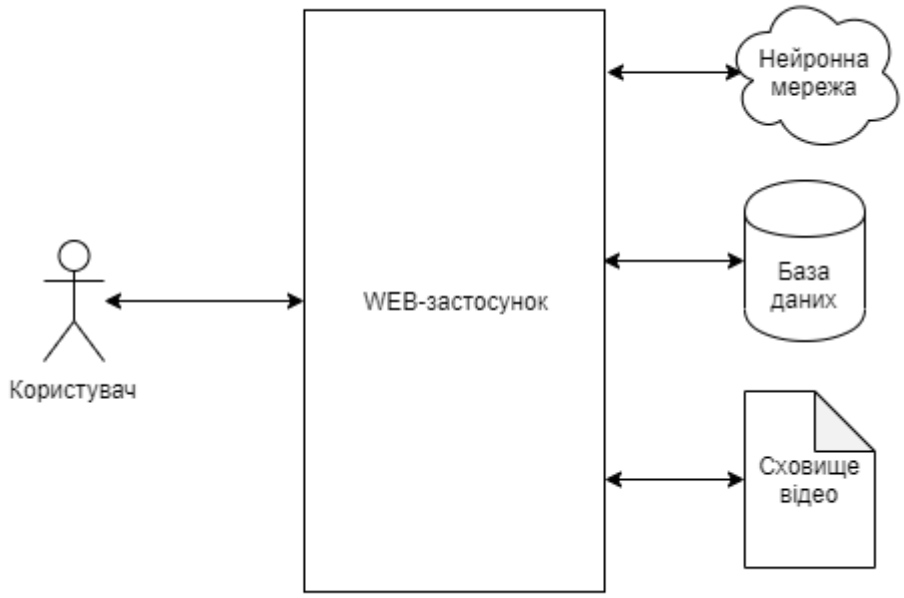


Рис. 3.1. Загальна схема структури розроблюваного програмного забезпечення

Отже «серцем» розроблюваного ПЗ є нейронна мережа, котра приймає відео і повертає припущення на рахунок оригінальності цього відеозапису. Проте користувач не має доступу до неї напряму. Усі операції з нейронною мережею відбуваються через так звану зовнішню систему – це WEB-застосунок, котрий має зв’язок із нейронною мережею, базою даних і сховищем відео, а також

використовується кінцевим користувачем. Дана WEB-аплікація має підсистему реєстрації, підсистему завантаження відео, підсистему взаємозв’язку з нейронною мережею.

Функції підсистеми реєстрації:

- Зареєструвати користувача;
- Авторизувати користувача;
- Дістати всю інформацію про дії користувача.

Функції підсистеми завантаження відео:

- Завантаження відео у систему;
- Порівняння завантаженого зараз відео із завантаженими відео в минулому;
- Збереження відео у сховищі;
- Оновлення інформації по даному відео для даного користувача;

Функції підсистеми взаємозв’язку з нейронною мережею:

- Відправка відеоматеріалу у нейронну мережу;
- Навчання нейронної мережі на основі даних кінцевих користувачів;
- Отримання відповіді про оригінальність відеоролику.

Усі ці підсистеми тісно зв’язані одна з одною проте можуть працювати окремо. Наприклад, якщо в даний момент часу нейронна мережа з якихось причин недоступна, то користувач все рівно буде мати доступ до WEB-сайту, де зможе увійти в свій аккаунт і побачити результати колишніх перевірок, або завантажити нове відео. Проте, нове відео буде залишатись у статусі «Оброблюється» оскільки нейронна мережа наразі не працює. Коли вона запрацює, то відео автоматично обробиться і користувач зможе побачити результат в особистому кабінеті.

Нейронна мережа працює окремо від WEB-додатку з декількох причин. По-перше, для усіх обрахувань мережі потрібно багато обчислювальної потужності, тому є логічним винести її на іншу машину. І по-друге, нейрона мережа розроблена використовуючи інший стек технологій із WEB-аплікацією,

тому зручніше запускати ці два додатки із різних місць, а спілкування між ними реалізувати за допомогою RESTful API.

З даною архітектурою постає питання розміщення всієї інфраструктури на окремих обчислювальних потужностях. Провідні компанії з цього напрямку є Amazon з AWS і Microsoft з Azure. Розглянемо плюси і недоліки кожного сервісу.

Цінові моделі як Azure, так і AWS пропонують оплату за структурою. AWS стягує з вас щогодини, тоді як Azure стягує плату за хвилину. Що стосується короткострокових планів підписки, Azure надає вам набагато більше гнучкості. У випадку певних послуг Azure, як правило, коштує дорожче, ніж AWS, коли архітектура починає масштабуватися.

Обчислення або сервіси обчислень є однією з основних послуг, що стосується хмарних обчислень. Оскільки велика кількість даних, що генеруються в ці дні, то завжди є потреба у швидших засобах обробки. Обчислювальні сервіси гарантують, що ви можете нерегулярно створювати екземпляри та миттєво збільшувати кількість примірників. AWS та Azure мають послуги, що задовольняють ці потреби.

AWS має такі сервіси, як EC2, Elastic Beanstalk, AWS Lambda, ECS і т.д. Однак, порівнюючи вартість, сервіси Microsoft, як правило, стають дорожчими в міру збільшення розміру. Якщо ви розглядаєте екземпляр з 256 ГБ оперативної пам’яті та 64vPCU, AWS стягуватиме з вас $3,20 за годину, тоді як Azure стягуватиме близько $6,76 за годину.

І AWS, і Azure надають тривалі та надійні послуги зберігання. AWS має такі послуги, як AWS S3, EBS та Glacier, тоді як служби зберігання Azure мають Blob Storage, Disk Storage та Standard Archive.

AWS S3 забезпечує високу доступність та автоматичну реплікацію в різних регіонах. Якщо мова йде про тимчасове зберігання в AWS, воно починає функціонувати кожного разу, коли екземпляр починається та зупиняється. Після закінчення він забезпечує блокове зберігання, схоже на жорсткі диски, і його можна приєднати до будь-якого екземпляра EC2 або зберігати окремо.

Azure використовує тимчасові сховища та веб-сторінки по об'єму VM. Azure має опцію Block Storage як аналог S3 в AWS. Крім того, Azure також пропонує два види їх зберігання, холодне та гаряче зберігання.

З точки зору довговічності, AWS має Amazon RDS, тоді як у Azure є база даних Azure SQL Server. Amazon RDS підтримує різні двигуни бази даних, такі як MariaDB, Amazon Aurora, MySQL, Microsoft SQL, PostgreSQL та Oracle, тоді як, коли мова заходить про Azure, база даних SQL Server базується на SQL, як підказує назва.

З точки зору інтерфейсу, у Azure він дружніший і плавніший, тоді як AWS забезпечує краще забезпечення більшої кількості екземплярів. Як видно, обидва інструменти мають свої особливості, якими можна похвалитися. Якщо говорити про доступність цих служб, вони досить рівноцінні, коли вони мають послуги для аналітики та Big Data. AWS має EMR, тоді як Azure має HD Insights для того ж. Azure також пропонує Cortana Intelligence Suite, який постачається з Hadoop, Spark, Storm та HBase.

Amazon Virtual Private Cloud (VPC) дозволяє створити ізольовані мережі під парасолькою Cloud. Це дозволяє користувачам створювати підмережі, таблиці маршрутів, приватні діапазони IP-адрес та мережеві шлюзи.

Віртуальна мережа Microsoft Azure як аналог VPC дозволяє вам робити все, що робить VPC. У обох постачальників є рішення про те, щоб також розповсюдити центр обробки даних у приміщенні на варіанти хмари та брандмауера.

Раніше встановлено, що AWS надає більш зрілі пропозиції Big Data та аналітики. У своєму арсеналі має різні сервіси, які охоплюють такі домени, як IoT, розробка мобільних додатків або створення обчислювального середовища залежно від потреб. Вони також пропонують підтримку Docker.

Microsoft тримається на рівні і може піти на крок далі, оскільки пропонує підтримку Hadoop з такими послугами, як Azure HDInsight. Варто звернути

увагу, що Windows Server 2016 забезпечує інтеграцію з Docker як для контейнерів Windows, так і для контейнерів Hyper-V.

Обидві хмарні платформи, як видно вище, демонструють потужні можливості, і важко вибрати чіткого переможця. Azure чудово підходить, коли мова йде про Hybrid Cloud та інтеграцію з пакетом продуктів Microsoft, тоді як AWS має більшу гнучкість та додаткові можливості. Отже Amazon Web Services являється передовим на сьогоднішній день провідником подібних послуг, тому для розроблюваного програмного забезпечення використано саме сервіси Amazon. Розглянемо конкретні сервіси Amazon AWS, котрі знадобляться для розроблюваної системи.

Усі системи розроблюваного продукту працюють під управлінням Amazon Web Services (Веб Сервіси Amazon). Це комерційна загальнодоступна хмара, що підтримується та розробляється Amazon з 2006 року. Вона надає абонентам послуги як за інфраструктурною моделлю (віртуальні сервери, ресурси для зберігання), так і за рівнем платформи (хмарні бази даних, хмарне проміжне програмне забезпечення, хмарні обчислення без серверів, засоби розробки).

Значною мірою (поряд із хмарною платформою Google) це вплинуло на формування концепції хмарних обчислень загалом і визначило основні напрямки розвитку моделі публічного розгортання. Тривалий час це була найбільша у світі публічна хмара за рівнем доходів, у другій половині 2010-х років програла Microsoft Azure за цим показником, зберігаючи домінування в сегментах інфраструктури та платформних послуг. Станом на 2017 рік річні доходи AWS перевищили 20 мільярдів доларів, що становить близько 11,5% доходів Amazon.

Він був офіційно запущений 14 березня 2006 року в рамках трьох сервісів: зберігання хмарних файлів Amazon S3, служби черги Amazon SQS та служб прокату обчислювальних можливостей Amazon EC2. У той же час компанія розпочала перші експерименти із надання таких послуг у липні 2002 року, наприкінці 2003 року Кріс Пінкхем та Бенджамін Блек розробили технічну та комерційну концепцію майбутнього AWS, яка передбачає використання таку ж

обчислювальну інфраструктуру як основу та програмні рішення, як у роздрібному бізнесі Amazon, а в листопаді 2004 р. Amazon SQS був протестований.

У грудні 2007 року була запущена перша хмарна СУБД, SimpleDB; через рік була запущена мережа доставки вмісту Amazon Cloud Front. У 2009 році з'явилися служби доступу до кластерів Hadoop (Elastic MapReduce) та реляційних СУБД (Amazon RDS). У 2012 році було запущено DynamoDB - хмарну СУБД NoSQL, RedShift - хмарну систему масового паралельного управління реляційними базами даних та систему довготривалого зберігання Amazon Glacier. Серверна обчислювальна платформа AWS Lambda була впроваджена в 2014 році, і серед нових великих запусків другої половини 2010-х, реляційна хмарна СУБД, сумісна з MySQL і PostgreSQL, DBMS Aurora та сервіс Elastic Kubernetes - це сервіс доступу до платформи контейнеризації на базі Kubernetes. До 2017 року загальна кількість хмарних сервісів перевищила 90.

Хмарні СУБД різних категорій широко представлені у хмарі. Серед доступних NoSQL-систем - Amazon SimpleDB, DynamoDB, резидентна СУБД ElastiCache, графічна СУБД Neptune. У рамках Служби реляційних баз даних Amazon (RDS) абоненти можуть розгортати хмарні бази даних, що працюють із популярними реляційними базами даних - MySQL, Oracle Database, Microsoft SQL Server та PostgreSQL, а масштабований реляційний бази даних Amazon Aurora, сумісний з MySQL та PostgreSQL, також є в наявності. ParAccel - аналітична система управління масовою паралельною реляційною базою даних, адаптована до хмарної інфраструктури, доступна під торговою маркою Amazon Redshift.

Сервіс Amazon Athena дозволяє аналізувати дані в Amazon S3 за допомогою стандартного SQL, і він не вимагає виділених обчислювальних потужностей, а абоненти платять лише за дані, прочитані в рамках виконаних запитів. Послуга Elastic MapReduce дозволяє абонентам створювати кластери Hadoop, оснащені відповідною екосистемою продуктів Big Data (включаючи

Spark, Hive, HBase, Presto). Інструмент QuickSight надає абонентам можливість візуально аналізувати дані, розміщені на AWS. Служба Amazon Elasticsearch надає хмарний доступ до стеку з пошукових систем Elasticsearch та Kibana. Amazon Machine Learning надає підписникам доступ до інструментів машинного навчання.

Серед сервісів середнього програмного забезпечення - брокер повідомлень Amazon Kinesis (аналогічний за можливостями Apache Kafka), служба черги SQS та служба сповіщення SNS.

Інструмент розгортання додатків у безхмарній обчислювальній парадигмі - AWS Lambda; Служба Elastic Kubernetes надає можливість розгортати програми в контейнерній інфраструктурі під управлінням Kubernetes.

WEB-додаток за реалізацією являє собою backend і frontend частину. Backend частина додатку розміщається на сервісі AWS EKS. Даний сервіс дає змогу запустити Kubernetes кластер на серверах Amazon без складної інсталяції. До плюсів такого підходу можна віднести надійність роботи додатку, оскільки Kubernetes автоматично стежить за стабільністю роботи аплікації і перезапускає її, якщо це треба. Також, Kubernetes дозволяє автоматично підіймати потужність додатку горизонтально масштабуючи його. Наприклад, можна виставити 100 запитів у секунду як поріг для розвертання додаткових Kubernetes Pods (далі – Под), тоді при навантаженні на додаток вище ніж 100 запитів у секунду, Kubernetes автоматично запустить ще один або декілька Подів, налаштувавши Loadbalancer (балансер навантаження) на розпаралелювання запитів на різні Поди, таким чином кожен з них буде функціонувати нормально і швидкість додатку не погіршиться.

Як і багато інших складних продуктів, Kubernetes впроваджує низку конкретних понять та понять у межах своєї екосистеми.

Вузол - це окрема фізична або віртуальна машина, на якій розгорнуті та запущені контейнери додатків. Кожен вузол кластера містить служби для

запуску програм у контейнерах (наприклад, Docker), а також компоненти, призначені для централізованого управління вузлом.

Под (від англійского pod) – основний блок для управління та запуску програм, один або кілька контейнерів, які гарантовано працюють на одному вузлі, забезпечується обмін ресурсами, міжпроцесорне спілкування та унікальна всередині кластера, IP-адреса. Останнє дозволяє додаткам, розгорнутим на поді, використовувати фіксовані та заздалегідь задані номери портів без ризику конфлікту. Струми можна безпосередньо керувати за допомогою API Kubernetes або ж вони можуть бути передані контролеру.

Том – це спільний ресурс сховища для спільного використання з контейнерів, розміщених в одному поді.

Усі об'єкти управління (вузли, поди, контейнери) у Kubernetes позначені мітками (ярликом), селекторами міток (селектором міток) є запити, що дозволяють отримати посилання на об'єкти, які відповідають деяким міткам; теги та селектори - головний механізм Kubernetes, який дозволяє вибрати, який об’єкт використовувати для запитуваної операції.

Сервіс в Kubernetes – це сукупність логічно пов’язаних наборів подів та політик доступу до них. Наприклад, сервіс може відповідати одному з шарів програмного забезпечення, розробленому відповідно до принципів багатошарової архітектури програмного забезпечення. Набір подів, що відповідають службі, отримують шляхом виконання селектора відповідної мітки.

Kubernetes надає функції виявлення служб та маршрутизації на вимогу, зокрема, система може перепризначити IP-адресу та доменне ім’я служби, необхідні для доступу до сервісу до різних подів. Це забезпечує збалансування навантаження між подами, мітки яких відповідають службі в стилі Round Robin DNS, а також правильну роботу в тому випадку, якщо один із вузлів кластера вийшов з ладу і розміщені на ньому поди автоматично переходять на інший . За замовчуванням послуга доступна всередині кластера, керованого Kubernetes, наприклад, резервні поди згруповані для забезпечення балансування

навантаження, і фронтенд надається в цій формі, але він також може бути налаштований для доступу до включених подів ззовні , як єдиний фронт.

Контролер – це процес, який контролює стан кластера, намагаючись довести його від фактичного до потрібного; він робить це за допомогою набору подів, який визначається за допомогою селекторів міток, що входять до визначення контролера. Виконання контролера забезпечується компонентом Kubernetes Controller Manager. Один з типів контролерів, найвідоміший – це контролер реплікації (Replication Controller), який забезпечує масштабування, запустивши вказану кількість копій пода в кластері. Він також забезпечує запуск нових екземплярів подів, якщо вузол, на якому працює под, яким керує цей контролер, виходить з ладу. Інші контролери, що входять до основної системи Kubernetes, включають контролер DaemonSet, який дає змогу запускати поди на кожній машині (або підмножині машин) та контролер роботи запускати поди, які виконуються до завершення, наприклад, як частина пакетної роботи.

Оператори - це спеціалізований тип програмного забезпечення Kubernetes, призначений для включення в кластер служб, що підтримують їх стан між стаціонарними виконаннями, такими як СУБД, системи моніторингу чи кешування. Мета операторів – надати можливість керувати стаціонарними програмами в кластері Kubernetes прозорим способом та приховати деталі їхніх налаштувань від основного процесу управління кластером Kubernetes.

Frontend частина застосунку являє собою Single Page Application, котрий розміщується на Amazon S3 Bucket, і коли користувач переходить на розроблюваний WEB-сайт то, за допомогою Amazon CloudFront, клієнт викачується на машину користувача і далі працює в локальному режимі, посилаючи запити на Backend. Коректне доменне ім’я налаштовується за допомогою Amazon CDN (Custom Domain Name), котрий співпрацює с CloudFront’ом.

Amazon CloudFront - це зручна для розробників глобальна мережа доставки вмісту (CDN), яка забезпечує швидку, безпечну передачу даних, відео, додатків

та API клієнтам у всьому світі з низькою затримкою та високою швидкістю. CloudFront інтегрований з AWS: його фізичні місця безпосередньо підключені до глобальної інфраструктури AWS, а програмне забезпечення ефективно працює з іншими службами, включаючи AWS Shield для нейтралізації DDoS-атак, Amazon S3, Elastic Load Balancing або Amazon EC2 як вихідних серверів для додатків, як а також Lambda@Edge для запуску спеціального коду в безпосередній близькості від кінцевих користувачів та налаштування вмісту. Використовуючи такі сервіси AWS, як Amazon S3, Amazon EC2 або Elastic Load балансування, вам не доведеться платити за передачу даних між цими службами та CloudFront.

Сховище відео це підсистема, куди завантажуються відеозаписи та зберігаються там доки не будуть перевірені за допомогою нейронної мережі. Після того як відеозапис пройде перевірку, він автоматично видаляється а інформація про оригінальність відео записується у базу даних. Також, якщо користувач при завантаженні відео заздалегідь знає відео підробка чи ні, то він може це помітити на сайті і тоді нейронна мережа спочатку дасть своє припущення, а потім використає це відео для поповнення датасету для навчання. Таким чином нейронна мережа має змогу навчатись не тільки на вихідному датасеті, а і за допомогою відеозаписів користувачів.

В якості сховища відео використовується Amazon S3 Bucket. Amazon S3 або Amazon Simple Storage Service – це сервіс, запропонований Amazon Web Services (AWS), який забезпечує зберігання об’єктів через інтерфейс веб-сервісу. Amazon S3 використовує ту саму масштабовану інфраструктуру зберігання, яку використовує Amazon.com для запуску глобальної мережі електронної комерції. Amazon S3 можна використовувати для зберігання будь-якого типу об’єктів, що дозволяє використовувати такі засоби, як зберігання для інтернет-додатків, резервне копіювання та відновлення, відновлення після аварій, архіви даних, озера даних для аналітики та гібридне хмарне зберігання. У своїй угоді про рівень обслуговування Amazon S3 гарантує 99,9% часу роботи, що працює

менше 43 хвилин простою на місяць. AWS запустила Amazon S3 у США 14 березня 2006 року, потім у Європі у листопаді 2007 року.

Хоча Amazon Web Services (AWS) публічно не надає подробиці технічного дизайну S3, Amazon S3 управляє даними з архітектурою об'єктів зберігання, яка має на меті забезпечити масштабованість, високу доступність та низьку затримку з 99,999999999% та міцністю від 99,95% до 99,99% (хоча не існує угоди про рівень обслуговування щодо довговічності).

Основними одиницями зберігання Amazon S3 є об'єкти, які організовані у відра. Кожен об'єкт ідентифікується за допомогою унікального призначеного користувачем ключа. Ведрами можна керувати за допомогою консолі, наданої Amazon S3, програмно за допомогою AWS SDK або за допомогою інтерфейсу програмування (API) програми Amazon S3 REST. Об'єктами можна керувати за допомогою AWS SDK або за допомогою API REST Amazon S3 і може бути розміром до п'яти терабайт з двома кілобайт метаданих. Крім того, об’єкти можна завантажувати за допомогою інтерфейсу HTTP GET та протоколу BitTorrent.

Запити дозволені за допомогою списку контролю доступу, пов’язаного з кожним відрізком об'єкта, та підтримки версії, яка за умовчанням вимкнена. Зауважте, що оскільки відра зазвичай мають розміри всієї файлової системи в інших системах, ця схема контролю доступу є грубозернистою, тобто ви не можете мати унікальні засоби контролю доступу, пов'язані з окремими файлами. Імена та ключі відра вибираються таким чином, щоб об’єкти були адресовані за допомогою URL-адрес HTTP:

- http://s3.amazonaws.com/bucket/key
- https://s3.amazonaws.com/bucket/key
- http://s3-region.amazonaws.com/bucket/key
- http://bucket.s3.amazonaws.com/key

В якості бази даних виступає Amazon DynamoDB. Amazon DynamoDB - це повністю керований фірмовий сервіс баз даних NoSQL, який підтримує

структуру даних ключових значень та документів і пропонується Amazon.com як частина портфоліо Amazon Web Services. DynamoDB відкриває аналогічну модель даних та отримує свою назву від Dynamo, але має іншу основу реалізації. "Динамо" створило багатопрофільний дизайн, який вимагав від клієнта вирішення конфліктів між версіями, а DynamoDB використовує синхронну реплікацію в декількох центрах обробки даних для високої довговічності та доступності. DynamoDB було оголошено CTO Amazon Werner Vogels 18 січня 2012 року і представляється як еволюція рішення Amazon SimpleDB.

DynamoDB відрізняється від інших служб Amazon тим, що дозволяє розробникам купувати послугу на основі пропускної здатності, а не на зберігання. Якщо ввімкнено автоматичне масштабування, то база даних буде масштабуватися автоматично. Крім того, адміністратори можуть запитувати зміни пропускної здатності, а DynamoDB поширюватиме дані та трафік на декілька серверів, використовуючи твердотілі накопичувачі, що забезпечує передбачувану продуктивність. Він пропонує інтеграцію з Hadoop через Elastic MapReduce.

У вересні 2013 року Amazon представила доступну версію для локальної розробки DynamoDB, щоб розробники могли тестувати додатки, підтримувані DynamoDB, на місцевому рівні.

Таблиця DynamoDB містить елементи, які мають атрибути, деякі з яких утворюють первинний ключ. У реляційних системах, однак, елемент містить кожен атрибут таблиці (або жонглює значеннями "null" та "unknown" за їх відсутності), елементи DynamoDB не мають схеми. Єдиний виняток: при створенні таблиці розробник вказує первинний ключ, і таблиця вимагає цього ключа для кожного елемента в ньому. Первинні ключі повинні бути скалярними (рядки, числа чи двійкові) і можуть мати одну з двох форм. Первинний ключ з одним атрибутом відомий як "ключ розділу" таблиці, який визначає розділ, який елемент має таким чином, ідеальний ключ розділу має рівномірне розподіл за своїм діапазоном. Первинний ключ також може містити другий атрибут, який

DynamoDB називає таблицею "сортування ключем". У цьому випадку ключі розділів не повинні бути унікальними; вони поєднуються з клавішами сортування, щоб зробити унікальний ідентифікатор для кожного елемента. Ключ розділу все ще використовується для визначення того, в якому розділі зберігається елемент, але в кожному розділі елементи сортуються за ключем сортування.

У реляційній моделі індекси, як правило, служать "допоміжною" структурою даних для доповнення таблиці. Вони дозволяють СУБД оптимізувати запити під капотом, і вони не покращують функціональність запитів. У DynamoDB немає оптимізатора запитів, а індекс – це просто інша таблиця з іншим ключем (або двома), що сидить поруч з оригіналом. Коли розробник створює індекс, він створює нову копію своїх даних, але копіюються лише ті поля, які вона вказує (як мінімум поля, які вона індексує, та первинний ключ оригінальної таблиці).

Користувачі DynamoDB видають запити безпосередньо на свої індекси. Є два типи індексів. Глобальний вторинний індекс містить ключ розділу (і необов'язковий ключ сортування), який відрізняється від ключа розділу оригінальної таблиці. Локальний вторинний індекс містить той самий ключ розділу, що і початкова таблиця, але інший ключ сортування. Обидва індекси вводять абсолютно нову функціональність запитів до бази даних DynamoDB, дозволяючи запити на нових клавішах. Як і у системах управління реляційними базами даних, DynamoDB оновлює індекси автоматично при додаванні / оновленні / видаленні, тому ви повинні бути обережними при їх створенні або ризикувати сповільнення важкої бази даних для запису з низкою оновлень індексу.

DynamoDB використовує JSON для свого синтаксису через всю його повсюдність. Для дії таблиці створення потрібні лише три аргументи: TableName, KeySchema – список, що містить ключ розділу та необов'язковий ключ сортування та AttributeDefinitions – список атрибутів, які слід визначити,

які повинна принаймні містити визначення для атрибутів, що використовуються як розділи та ключі сортування. У той час як реляційні бази даних пропонують надійні мови запитів, DynamoDB пропонує операції Put, Get, Update та Delete. Запити розміщення містять атрибут TableName та атрибут Item, який складається з усіх атрибутів та значень, які має елемент. Запит на оновлення відповідає тому ж синтаксису. Аналогічно, щоб отримати або видалити елемент, треба просто вказати назву таблиці та ключ.

3.2 Розробка нейронної мережі

У цьому підрозділі описуються створення нейронної мережі, а також вибір найкращих для цього інструментів і алгоритмів. Для розпізнавання Deep Fake відеозаписів використовується розроблена нейронна мережа. Спочатку, вона має розпізнати обличчя, далі виділити одиниці дії обличчя з відео і після цього, на основі попереднього навчання, зробити припущення про оригінальність даного відеозапису. Існує декілька відомих алгоритмів розпізнавання обличчя, розглянемо їх більш детально.

Метод гнучкого порівняння на графах (Elastic graph matching). Суть методу зводиться до еластичного порівнянні графів, що описують зображення осіб. Особи представлені у вигляді графів зі зваженими вершинами і ребрами. На етапі розпізнавання один з графів – еталонний – залишається незмінним, в той час як інший деформується з метою найкращої підгонки до першого. У подібних системах розпізнавання графи можуть являти собою як прямокутну решітку, так і структуру, утворену характерними (антропометричними) точками особи (рис. 3.2).

У вершинах графа обчислюються значення ознак, найчастіше використовують комплексні значення фільтрів Габора або їх упорядкованих наборів – Габоровскіх вейвлет (строї Габора), які обчислюються в деякій

локальній області вершини графа локально шляхом згортки значень яскравості пікселів з фільтрами Габора.

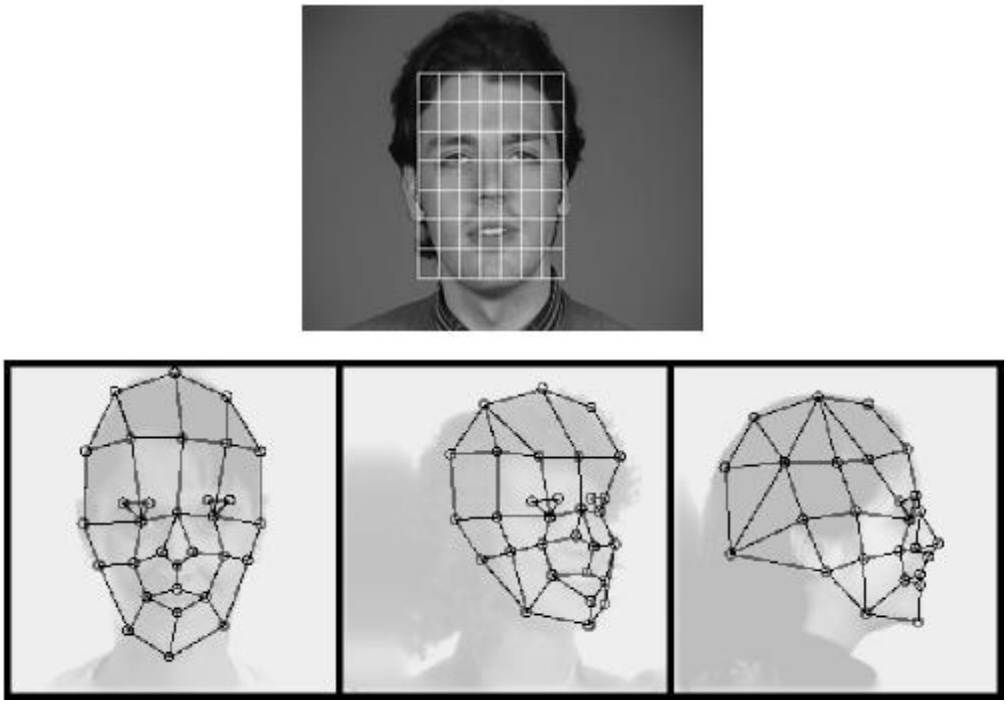


Рис. 3.2. Приклад структури графа для розпізнавання обличчя: згори регулярна решітка, знизу – граф на основі антропометричних точок обличчя

Ребра графа зважуються відстанями між суміжними вершинами. Різниця (відстань, дискримінаційна характеристика) між двома графами обчислюється за допомогою деякої цінової функції деформації, що враховує як відмінність між значеннями ознак, обчисленими в вершинах, так і ступінь деформації ребер графа.

Деформація графа відбувається шляхом зсуву кожної з його вершин на деяку відстань в певних напрямках щодо її вихідного розташування і вибору такої її позиції, при якій різниця між значеннями ознак (відгуків фільтрів Габора) в вершині деформованого графа і відповідної їй вершині еталонного графа буде мінімальною. Дана операція виконується по черзі для всіх вершин графа до тих пір, поки не буде досягнуто найменше сумарне відмінність між ознаками деформованого і еталонного графів. Значення цінової функції деформації при

такому положенні, що деформується графа і буде мірою відмінності між вхідним зображенням обличчя і еталонним графом. Дана «релаксаційна» процедура деформації повинна виконуватися для всіх еталонних осіб, закладених в базу даних системи. Результат розпізнавання системи - еталон з найкращим значенням цінової функції деформації (рис. 3.3).

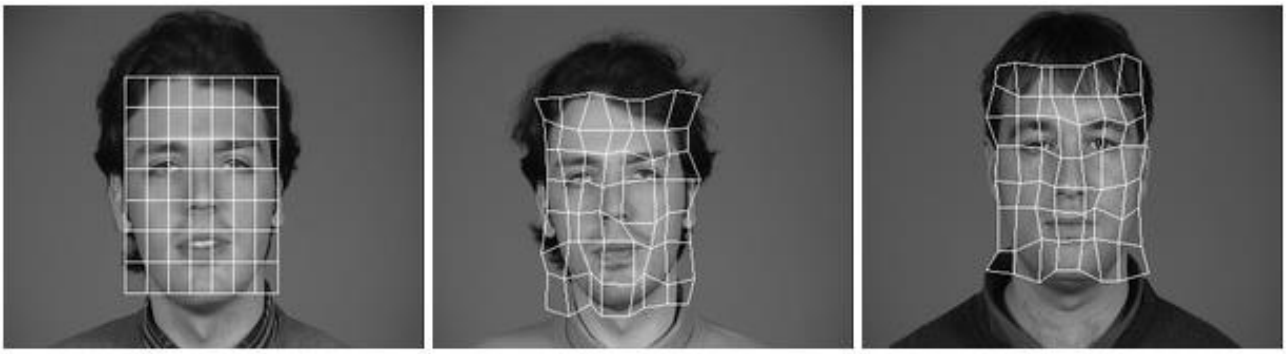


Рис. 3.3. Приклад деформації графа у вигляді регулярної решітки

В окремих публікаціях вказується 95 – 97 % ефективність розпізнавання навіть при наявності різних емоційних висловах і зміні ракурсу особи до 15 градусів. Однак розробники систем еластичного порівняння на графах посилаються на високу обчислювальну вартість даного підходу. Наприклад, для порівняння вхідного зображення особи з 87 еталонними витрачалося приблизно 25 секунд при роботі на паралельній ЕОМ з 23 транспьютерами. В інших публікаціях з даної тематики час або не вказується, або говориться, що воно велике.

Недоліки: висока обчислювальна складність процедури розпізнавання. Низька технологічність при запам'ятовуванні нових еталонів. Лінійна залежність часу роботи від розміру бази даних осіб.

Нейронні мережі. В даний час існує близько десятка різновидів нейронних мереж (НМ). Одним з найбільш широко використовуваних варіантів є мережа, побудована на багатошаровому перцептроні, яка дозволяє класифікувати подане

на вхід зображення/сигнал відповідно до попередніх налаштувань/навчання мережі.

Навчаються нейронні мережі на наборі навчальних прикладів. Суть навчання зводиться до налаштування ваг міжнейронних зв'язків в процесі рішення оптимізаційної задачі методом градієнтного спуску. В процесі навчання НС відбувається автоматичне вилучення ключових ознак, визначення їх важливості та побудова взаємозв'язків між ними. Передбачається, що навчена НС зможе застосувати досвід, отриманий в процесі навчання, на невідомі образи за рахунок узагальнюючих здібностей.

Найкращі результати в області розпізнавання осіб (за результатами аналізу публікацій) показала Convolutional Neural Network або згорткова нейронна мережа (ЗНМ), яка є логічним розвитком ідей таких архітектур НМ як когнітрона і неокогнітрона (рис. 3.4). Успіх обумовлений можливістю обліку двовимірної топології зображення, на відміну від багатошарового перцептрона.

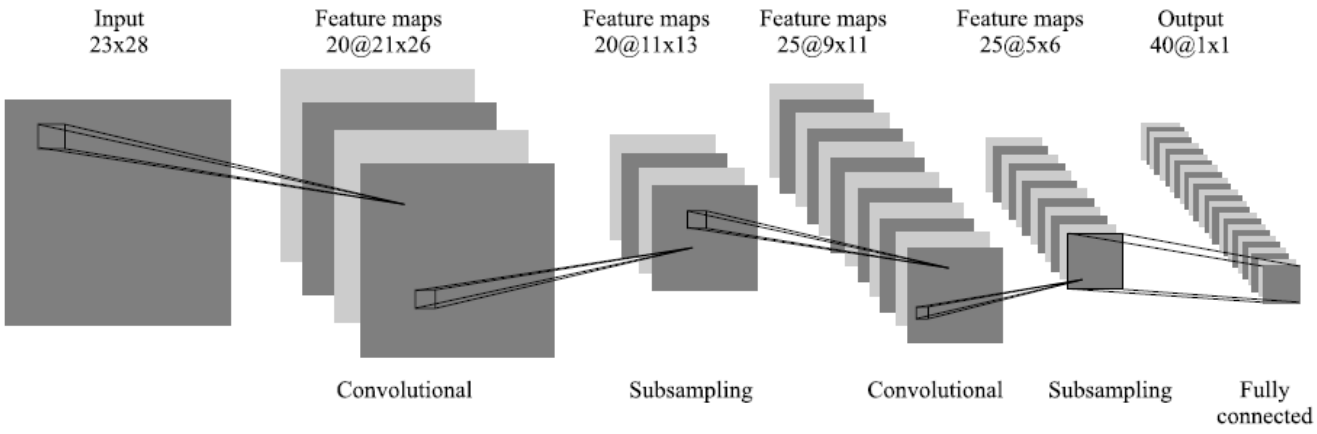


Рис. 3.4. Схематичне зображення архітектури згорткової нейронної мережі

Відмінними рисами ЗНМ є локальні рецепторні поля (забезпечують локальну двовимірну зв'язність нейронів), загальні ваги (забезпечують детектування деяких рис в будь-якому місці зображення) і ієрархічна організація з просторовими семплінгом (spatial subsampling). Завдяки цим нововведенням

ЗНМ забезпечує часткову стійкість до змін масштабу, зсувів, поворотам, зміні ракурсу і іншим спотворень.

Тестування ЗНМ на базі даних ORL, що містить зображення осіб з невеликими змінами освітлення, масштабу, просторових поворотів, положення і різними емоціями, показало 96% точність розпізнавання.

Свій розвиток ЗНМ отримали в розробці DeepFace (рис. 3.5), яку придбав Facebook для розпізнавання осіб користувачів своєї соцмережі. Всі особливості архітектури носять закритий характер.

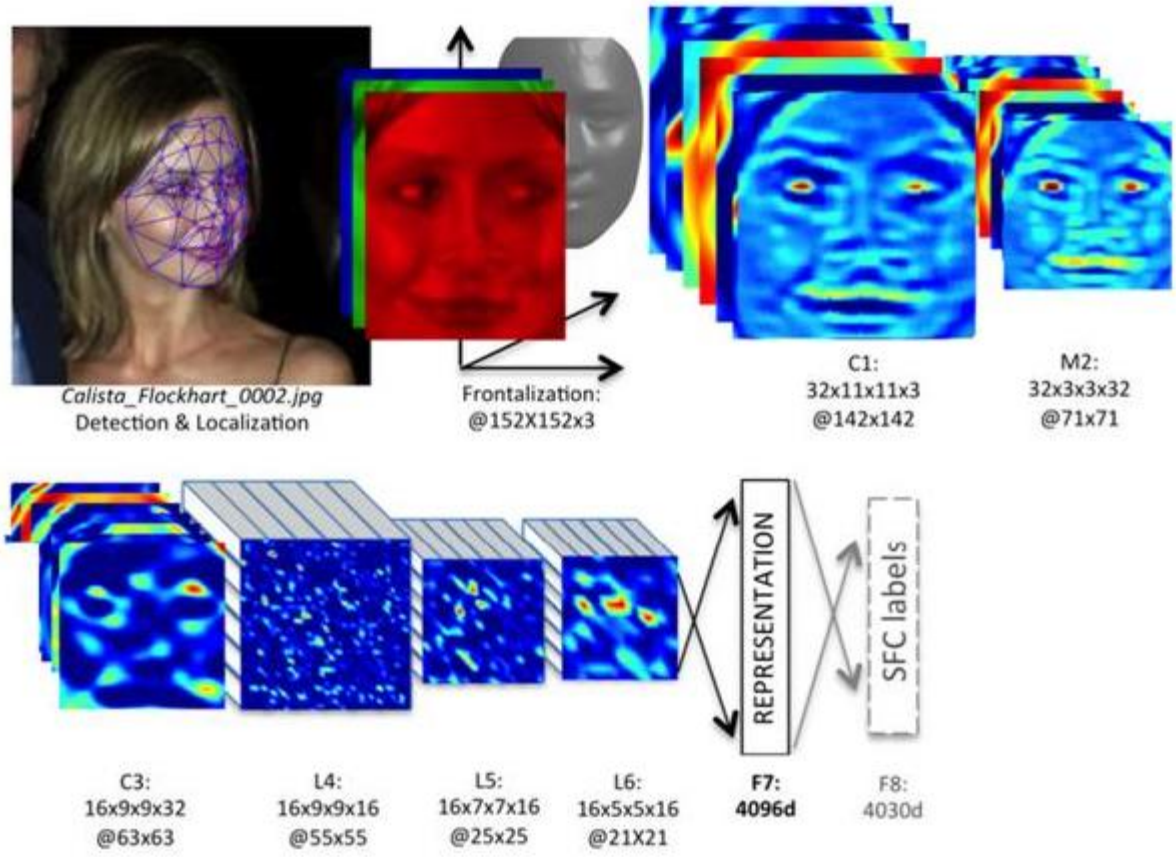


Рис. 3.5. Принцип роботи Deep Face

Недоліки нейронних мереж: додавання нової еталонної особи в базу даних вимагає повного перенавчання мережі на всьому наявному наборі (досить тривала процедура, в залежності від розміру вибірки від 1 години до декількох днів). Проблеми математичного характеру, пов'язані з навчанням: потрапляння в локальний оптимум, вибір оптимального кроку оптимізації, перенавчання і т.д.

Важко формалізується етап вибору архітектури мережі (кількість нейронів, шарів, характер зв'язків).

Прихована марковська модель (ПММ, англ. HMM). Одним із статистичних методів розпізнавання осіб є приховані марковскі моделі з дискретним часом. ПММ використовують статистичні властивості сигналів і враховують безпосередньо їх просторові характеристики. Елементами моделі є: безліч прихованих станів, безліч спостережуваних станів, матриця перехідних ймовірностей, початкова ймовірність станів. Кожному відповідає своя марковська модель. При розпізнаванні об'єкта перевіряються згенеровані для заданої бази об'єктів марковські моделі і відшукується максимальна із спостережуваних ймовірність того, що послідовність спостережень для даного об'єкта згенерована відповідною моделлю.

На сьогоднішній день не вдалося знайти приклад комерційного застосування ПММ для розпізнавання осіб. Недоліки:

- необхідно підбирати параметри моделі для кожної бази даних;
- ПММ не володіє здатністю розрізняти, тобто алгоритм навчання тільки максимізує відгук кожного зображення на свою модель, але не мінімізує відгук на інші моделі.

Метод головних компонент або principal component analysis (PCA). Одним з найбільш відомих і опрацьованих є метод головних компонент (principal component analysis, PCA), заснований на перетворенні Карунена-Лоєва.

Спочатку метод головних компонент почав застосовуватися в статистиці для зниження простору ознак без істотної втрати інформації. У задачі розпізнавання осіб його застосовують головним чином для представлення зображення особи вектором малої розмірності (головних компонент), який порівнюється потім з еталонними векторами, закладеними в базу даних.

Головною метою методу головних компонент є значне зменшення розмірності простору ознак таким чином, щоб воно якомога краще описувало «типові» образи, що належать безлічі осіб. Використовуючи цей метод можна

виявити різні мінливості в навчальній вибірці зображень обличь і описати цю мінливість в базисі декількох ортогональних векторів, які називаються власними (eigenface).

Отриманий один раз на навчальній вибірці зображень обличь набір власних векторів використовується для кодування всіх інших зображень осіб, які представляються зваженої комбінацією цих власних векторів. Використовуючи обмежену кількість власних векторів можна отримати стислу апроксимацію вхідному зображенню особи, яку потім можна зберігати в базі даних у вигляді вектора коефіцієнтів, який служить одночасно ключем пошуку в базі даних осіб.

Суть методу головних компонент зводиться до наступного. Спочатку весь навчальний набір осіб перетвориться в одну загальну матрицю даних, де кожен рядок являє собою один екземпляр зображення особи, розкладеного в рядок. Всі особи навчального набору повинні бути приведені до одного розміру і з нормованими гістограмами.

Потім проводиться нормування даних і приведення рядків до 0-го середнього і 1-й дисперсії, обчислюється матриця коваріації. Для отриманої матриці коваріації вирішується завдання визначення власних значень і відповідних їм власних векторів (власні особи). Далі проводиться сортування власних векторів в порядку убування власних значень і залишають тільки перші k векторів за правилом:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i} > 0{,}95$$

Метод головних компонент добре зарекомендував себе в практичних додатках. Однак, в тих випадках, коли на зображенні особи присутні значні зміни в освітленості або виразі обличчя, ефективність методу значно падає. Вся справа в тому, що PCA вибирає підпростір з такою метою, щоб максимально апроксимувати вхідний набір даних, а не виконати дискримінацію між класами осіб.

В [12] було запропоновано вирішення цієї проблеми з використанням лінійного дискримінанту Фішера (в літературі зустрічається назва "Eigen-Fisher", "Fisherface", LDA). LDA вибирає лінійний підпростір, який максимізує відношення:

$$\frac{|\phi^T s_b \phi|}{|\phi^T s_w \phi|}$$

де $S_b = \sum_{i=1}^{m} N_i (\bar{X}_i - \bar{X})(\bar{X}_i - \bar{X})^T$

матриця міжкласового розкиду, і

$$S_w = \sum_{i=1}^{m} \sum_{x \in X_i} \left(X - \vec{X}_i\right)(X - \bar{X}_i)^T$$

матриця розкиду всередині класу; m - число класів в базі даних.

LDA відшукує проєкцію даних, при якій класи є максимально лінійно роздільні (рис. 3.6).

Для порівняння PCA шукає таку проєкцію даних, при якій буде максимізований розкид по всій базі даних осіб (без урахування класів). За результатами експериментів [12] в умовах сильного бакового і нижнього затінення зображень осіб Fisherface показав 95 % ефективність у порівнянні з 53 % Eigenface.

Відмінності методу головних компонент PCA від методу лінійного дискримінанту Фішера LDA:

- PCA здійснює зменшення розмірності, зберігаючи якомога більше дисперсій у просторі з високими розмірами.
- LDA здійснює зменшення розмірності, зберігаючи якомога більше дискримінаційної інформації класу.

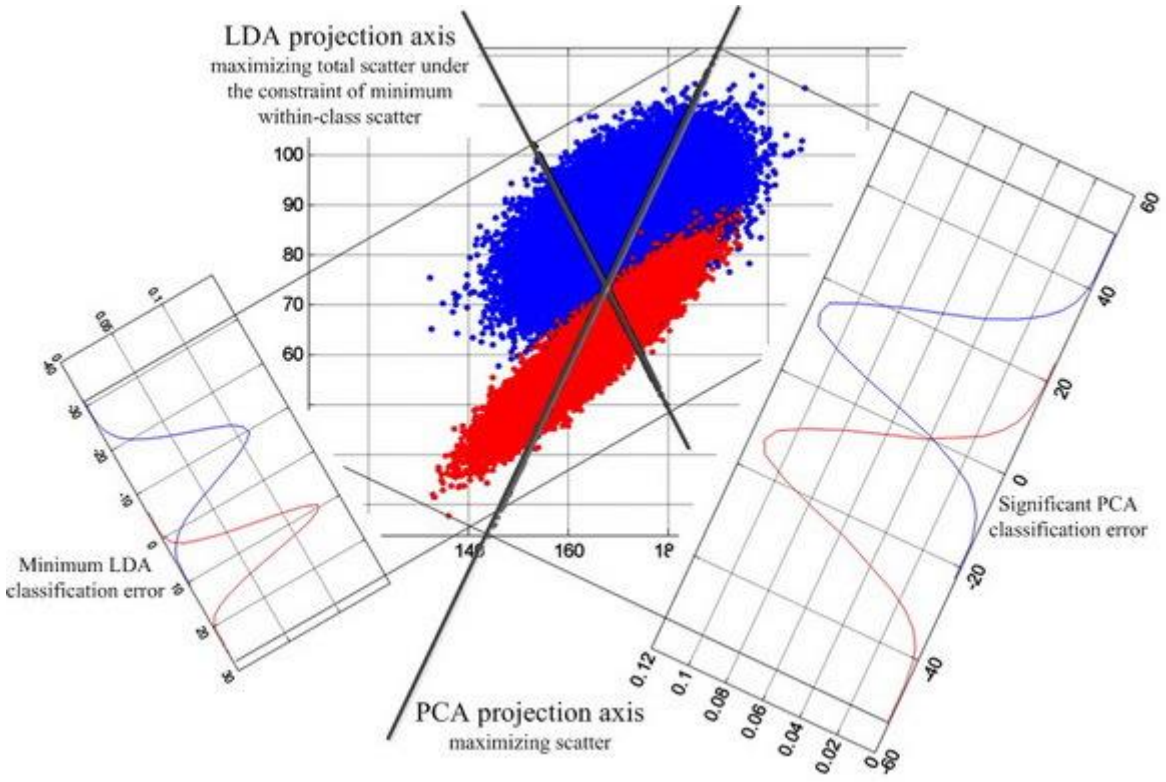


Рис. 3.6. Принципова різниця між формуванням проекції PCA і LDA

Моделі активного вигляду (Active Appearance Models). Це статистичні моделі зображень, які шляхом різного роду деформацій можуть бути підігнані під реальне зображення. Даний тип моделей в двовимірному варіанті було запропоновано Тімом Кутса і Крісом Тейлором в 1998 році [11]. Спочатку активні моделі зовнішнього вигляду застосовувалися для оцінки параметрів зображень облич.

Активна модель зовнішнього вигляду містить два типи параметрів: параметри, пов'язані з формою (параметри форми), і параметри, пов'язані зі статистичною моделлю пікселів зображення або текстурою (параметри зовнішнього вигляду). Перед використанням модель повинна бути навчена на безлічі заздалегідь розмічених зображень (рис. 3.7). Розмітка зображень виробляється вручну. Кожна мітка має свій номер і визначає характерну точку, яку повинна буде знаходити модель під час адаптації до нового зображення.

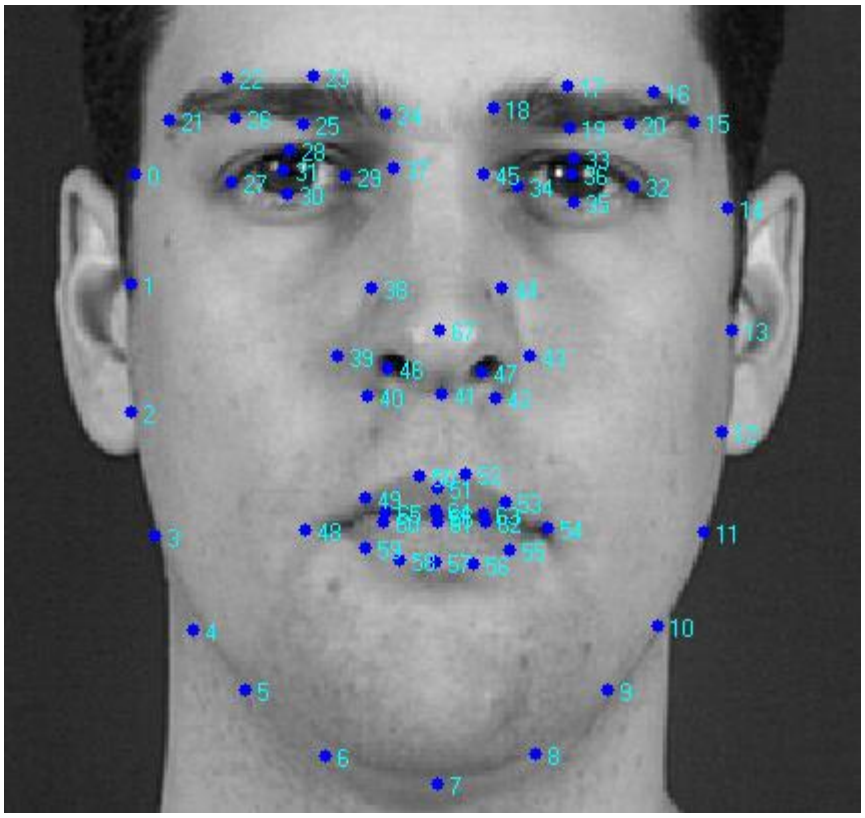


Рис. 3.7. Приклад розмітки зображення обличчя за допомогою 68 точок, які створюють форму AAM

Отже, для реалізації нейронної мережі по розпізнаванню Deep Fake відео обрано алгоритм згорткових нейронних мереж (ЗНМ), оскільки даний метод розпізнавання обличчя показує найкращі результати на однакових тестових датасетах, а також не потребує великої обчислювальної потужності.

Також, існує декілька різних підходів для розробки нейронної мережі. Можна створити її повністю з нуля, проте це дуже складно, є велика ймовірність помилитись і це займає багато часу. А можна використати вже існуючі інструменти, котрі зосередять увагу на комбінуванні необхідних алгоритмів і підходів, а не на низькорівневому програмуванні нейронної мережі. Такий підхід є набагато зручнішим, швидшим і потужнішим, адже усі алгоритми були створені і протестовані заздалегідь. Таким інструментом для розроблюваної системи є TensorFlow.

TensorFlow – відкрита програмна бібліотека для машинного навчання цілій низці задач, розроблена компанією Google для задоволення її потреб у системах, здатних будувати та тренувати нейронні мережі для виявлення та розшифровування образів та кореляцій, аналогічно до навчання й розуміння, які застосовують люди. Її наразі застосовують як для досліджень, так і для розробки продуктів Google, часто замінюючи на його ролі її закритого попередника, DistBelief. TensorFlow було початково розроблено командою Google Brain для внутрішнього використання в Google, поки її не було випущено під відкритою ліцензією Apache 2.0 9 листопада 2015 року.

Архітектура Tensorflow працює в трьох частинах:

- Попередня обробка даних;
- Побудування моделі;
- Тренування і оцінка моделі.

Бібліотека називається TensorFlow, оскільки вона приймає на вхід багатовимірний масив, також відомий як тензор. Ви можете побудувати своєрідну блок-схему операцій (граф), яку ви хочете виконати на цьому вході. Вхід надходить в один кінець, а потім він протікає через цю систему декількох операцій і виходить з іншого кінця як вихід. Розглянемо детальніше компоненти TensorFlow.

Тензор (Tensor). Назва TensorFlow прямо походить від основи: Тензора (Tensor). У Tensorflow всі обчислення включають тензори. Тензор - це вектор або матриця n-розмірів, що представляє всі типи даних. Усі значення в тензорі містять однаковий тип даних з відомою (або частково відомою) формою. Форма даних - це розмірність матриці або масиву.

Тензор може виникнути з вхідних даних або з результату обчислення. У TensorFlow всі операції проводяться всередині графа. Граф - це набір обчислень, який відбувається послідовно. Кожна операція називається op-вузлом і з'єднана між собою.

На графу окреслюються вузли та з'єднання між вузлами. Однак значення не відображаються. Грані вузлів – це тензор, тобто спосіб заповнити операцію даними.

Для витягування рухів обличчя та голови у відео використано інструмент аналізу обличчя з відкритим кодом OpenFace2. Ця бібліотека містить 2-D та 3-D орієнтири для обличчя, позу голови, очні погляди та одиничні дії для кожного кадру в даному відео. Приклад вилучених вимірювань показаний на рис. 3.8.

Рухи м’язів обличчя можна представити у вигляді одиниць дії обличчя (ОД). За допомогою OpenFace2 виділено 17 ОД: внутрішній підйом брови (ОД01), зовнішній підйом брови (ОД02), опущення брови (ОД04), підйом верхньої повіки (ОД05), підйом щоки (ОД06), стискання повіки (ОД07), морщення носа (ОД09), підйом верхньої губи (ОД10), піднімання куточка губи (ОД12), ямочка (ОД14), опускання куточка губи (ОД15), підйом підборіддя (ОД17), розтягування губ (ОД20), стискання губ (ОД23), розведення двох губ (ОД25), опускання щелепи (ОД26) та моргання очей (ОД45).

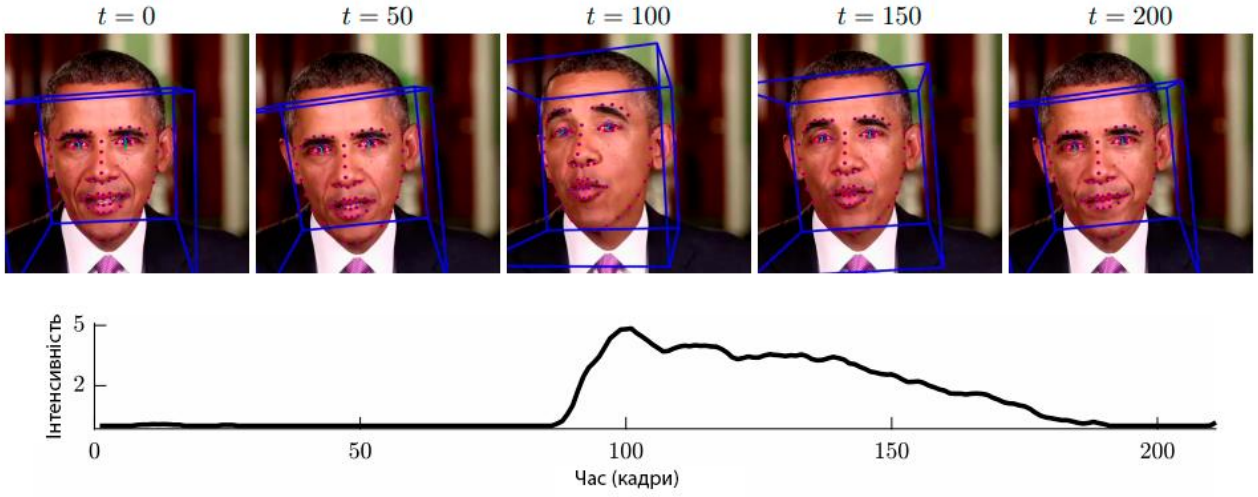


Рис. 3.8. П’ять кадрів з нанесеною інформацією від OpenFace та графік інтенсивності однієї одиниці дії обличчя ОД01 (піднімання брів), що виміряно на цьому відеозаписі

Розроблювана модель містить 16 ОД – ОД миготіння очей було усунене, оскільки воно недостатньо точне для поставлених цілей. Ці 16 ОД доповнені такими чотирма ознаками: (1) обертання голови навколо осі x (кивання); (2) обертання голови навколо осі z (нахил); (3) 3-D горизонтальна відстань між куточками рота (poth); та (4) 3-D вертикальна відстань між нижньою та верхньою губою (potv). Перша пара особливостей фіксує загальний рух голови (ми не вважаємо обертання навколо осі y (поворот) через відмінності, коли йде розмова безпосередньо з людиною та з великим натовпом). Друга пара цих функцій фіксує розтягнення рота (ОД27) та смоктання губ (ОД28), які не охоплюються нашими стандартними 16 ОД.

Використано кореляцію Пірсона для вимірювання лінійності між цими ознаками, щоб охарактеризувати особистий рух людини. Маючи в цілому 20 функцій обличчя / голови, обчислюємо кореляцію Пірсона між усіма 20 цими характеристиками, отримуючи 20C2 = (20 × 19) / 2 = 190 пар функцій у всіх 10-секундних відеокліпах, що перекриваються. Кожен 10-секундний відеокліп зводиться до характеристичного вектора розміром 190, який, як описано далі, потім використовується для класифікації відео як справжнього або підробленого.

З цими алгоритмами проведено навчання нейронної мережі. Для навчання використовувались два набори даних. Власноруч завантажені відео із YouTube і відкритий датасет Deep Fake відео від компанії Google.

На даному етапі залишається лише одна задача – зв’язати нейронну мережу із системою для кінцевого користувача. Окрім самої нейронної мережі створено невеликий додаток за допомогою мови Python, котрий слухає топік в Amazon SNS. Це сервіс від Amazon, котрий дозволяє зручно робити publisher/subscriber брокери повідомлень. Це працює наступним чином:

- Python додаток слухає нові повідомлення в Amazon SNS
- Завантажується відеозапис в порядку черги
- Відбувається навчання/припущення про оригінальність відео
- Відправляється зворотній запит на систему для кінцевого користувача

У SNS повідомлені зберігається посилання на S3 Bucket з потрібним відео. Таким чином Python додаток з нейронною мережею може легко завантажити відео, щоб виконувати з ним подальші дії. Доступ до S3 Bucket надається завдяки AWS IAM Policy, а завантаження відео здійснено за допомогою AWS SDK для мови Python.

Сервіс AWS Identity and Access Management (IAM) надає можливість безпечного управління доступом до сервісів та ресурсів AWS. Використовуючи IAM, можна створювати користувачів AWS і групи, керувати ними, а також використовувати дозволу, щоб надавати або забороняти доступ до ресурсів AWS.

AWS SDK це набір інструментів, які надають розробнику можливість звертатись до сервісів Amazon без необхідності створення запитів вручну. AWS SDK існує для багатьох мов програмування. В даному випадку використано для мови Python.

Отже, розроблено нейронну мережу за допомогою мови програмування Python і бібліотеки TensorFlow. Дана система побудована на основі згорткових нейроних мереж і використовує OpenFace2 для витягування рухів обличчя і голови. Проведено навчання нейронної мережі за допомогою двох вищеназваних датасетів. Також, розроблено Python додаток, який дозволяє інтегрувати нейронну мережу у сервіси AWS і зв’язати її із системою для кінцевого користувача.

3.3 Розробка системи для кінцевого користувача

Даний підрозділ описує інструменти для розробки під дану задачу, розкриває розроблені підсистеми, описує схему бази даних.

Для розробки системи для кінцевого користувача постає задача вибору підходящих інструментів, таких як мова програмування, відповідні фреймворки і бібліотека, база даних, допоміжні засоби і CI/CD сценарії. Серед мов програмування для цієї системи постав вибір між C# і Java. Ці дві мови дуже схожі одна на одну, проте є і значні відмінності.

Поява як Java, так і C #, тісно пов'язана з переходом від низькорівневих мов програмування, таких як мови програмування C++, до мов вищого рівня, які компілюються в байт-код. Байт-код можна запустити на віртуальній машині. З цим пов'язаний ряд переваг, в першу чергу, можливість написання коду, який буде зрозумілий людині і буде працювати на будь-якій апаратній архітектурі, на якій встановлена віртуальна машина. Якщо відкинути синтаксичні примхи в сторону, то не дивно, що ці два подібні між собою мови так популярні для розробників додатків. Ось кілька основних подібностей між C# і Java:

Безпека типів. Помилка типу виникає, коли тип даних одного об'єкта помилково призначається іншому об'єкту, створюючи ненавмисні побічні ефекти. І C#, і Java працюють на те, щоб гарантувати виявлення таких незаконних привидів під час компіляції. Якщо приведення не може бути застосоване до нового типу, тоді під час виконання такі винятки будуть видалені.

Прибирання сміття: На мовах нижчого рівня управління пам'яттю може бути стомлюючим, адже потрібно пам'ятати про те, що необхідно правильно видалити нові об'єкти, щоб звільнити ресурси. На C# і Java є вбудована прибирання сміття, яка допомагає запобігти витоку пам'яті шляхом видалення об'єктів, які більше не використовуються додатком. Витоку пам'яті все ще можуть виникати, але завдяки основам управління пам'яттю - це вже не ваша проблема.

Одиночне наслідування. Обидві мови підтримують одиночне наслідування – це означає, що існує тільки один шлях з будь-якого базового класу в будь-який з його похідних класів. Це обмежує ненавмисні побічні ефекти, які можуть

виникати при наявності декількох шляхів між декількома базовими класами і похідними класами. Diamond pattern – розповсюджений приклад цієї проблеми.

Інтерфейси. Інтерфейс являє собою абстрактний клас, де всі методи абстрактні. Абстрактним методом є той метод, який оголошений, але не містить подробиць його реалізації. Код, що визначає будь-які методи або властивості, певні інтерфейсом, має надаватися класом, який його реалізує. Це допомагає уникнути двозначності паттерна diamond, оскільки завжди ясно, який базовий клас реалізує даний похідний клас під час виконання. Результатом є чиста ієрархія лінійних класів одиночного наслідування в поєднанні з деякою універсальністю множинного спадкоємства. Фактично використання абстрактних класів є одним із способів множинного наслідування мов, які можуть подолати проблему паттерна diamond.

Важливо пам'ятати, що C# бере свій початок в бажанні Microsoft мати власну «Java-подібну» мову для платформи .NET. Оскільки C# не створювалася в вакуумі, нові функції були додані і налаштовані для вирішення проблем, з якими стикалися розробники Microsoft, коли вони спочатку намагалися створити свою платформу на Visual J++. У той же час співтовариство Java з відкритим вихідним кодом продовжувала зростати і між цими двома мовами розвивалася гонка технічних озброєнь. Ось деякі з основних відмінностей між C# і Java.

Windows чи open-source. Хоча існують реалізації з відкритим вихідним кодом, C# в основному використовується в розробці для платформ Microsoft – NET Framework CLR і є найбільш широко використовуваної реалізацією CLI. На іншому кінці спектру Java має величезну екосистему з відкритим вихідним кодом і у нього відкрилося друге дихання частково завдяки тому, що Google використовує JVM для Android.

Підтримка узагальнень (Generics): Generics покращує перевірку типів за допомогою компілятора, в основному видаляючи приведення з вихідного коду. В Java кошти узагальнень реалізуються з використанням стирань. Параметри загального типу «стираються», а при компіляції в байт-код додаються

приведення. C# також використовує узагальнення, інтегруючи його в CLI і надаючи інформацію про тип під час виконання, що дає невелике збільшення продуктивності.

Підтримка делегатів (покажчиків): У C # є делегати, які по суті є як методів, які можуть бути викликані не повідомляючи цільового об'єкта. Для досягнення такої ж функціональності в Java вам необхідно використовувати інтерфейс з одним методом або іншим способом обходу, який може зажадати нетривіального кількості додаткового коду, в залежності від програми.

Перевіряються виключення: Java розрізняє два типи винятків – checked і unchecked. C# вибрав більш мінімалістський підхід, маючи тільки один тип винятку. Хоча здатність ловити виключення може бути корисна, вона також може мати негативний вплив на масштабованість і контроль версій.

Поліморфізм: C# і Java використовують дуже різні підходи до поліморфізму. Java допускає поліморфізм за замовчуванням, C# же повинен викликати ключове слово «virtual» в базовому класі і ключове слово «override» в похідному класі.

Перерахування (Enums): в C# перерахування представляють собою прості списки іменованих констант, де базовий тип повинен бути цілим. Java являє перерахування більш глибоко, розглядаючи його як іменований екземпляр типу, що спрощує додавання користувацького поведінки окремих перерахуванням.

Отже, оскільки розроблюваний проєкт являється WEB-додатком, то вирішено використовувати Java, так як ця мова спрямована на WEB оточення, а C# в основному використовується для Windows додатків. Крім того, розроблюваний продукт працює за допомогою сервісів Amazon (AWS), а дані сервіси набагато краще пристосовані для роботи із мовою Java.

Проте мова програмування Java дуже рідко використовується без фреймворків, оскільки в такому разі розробка набуває дуже низькорівневого стану і сильно сповільнюється швидкість написання програми.

Фреймворк — це готовий до використання комплекс програмних рішень, включаючи дизайн, логіку та базову функціональність системи або підсистеми. Відповідно програмний фреймворк може містити в собі також допоміжні програми, деякі бібліотеки коду, скрипти та загалом все, що полегшує створення та поєднання різних компонентів великого програмного забезпечення чи швидке створення готового і не обов'язково об'ємного програмного продукту. Побудова кінцевого продукту відбувається, зазвичай, на базі єдиного API.

В якості фреймворку для мови Java обрано Spring, оскільки це найбільший і найсучасніший фреймворк з усіх існуючих. У цього інструмента мільйони користувачів по всьому світові, тому час на пошук і вирішення проблеми буде мінімізовано завдяки активній підтримці користувачів.

Spring Framework – це програмний каркас (фреймворк) з відкритим кодом та контейнери з підтримкою інверсії управління для платформи Java.

Основні особливості Spring Framework можуть бути використані будь-яким додатком Java, але є розширення для створення веб-додатків на платформі Java EE. Незважаючи на це, Spring Framework не нав'язує якоїсь конкретної моделі програмування, Spring Framework став популярним в спільноті Java як альтернатива, або навіть доповнення моделі Enterprise JavaBean (EJB).

В рамках даного підрозділу розроблено наступні підсистеми:

- Підсистема авторизації;
- Підсистема завантаження відео;
- Підсистема взаємодії з нейронною мережею;
- Підсистема роботи з базою даних;
- Підсистема прийому платежів.

Підсистема авторизації представляє собою модуль Java додатку, котрий відповідає за реєстрацію і авторизацію користувачів. Аутентифікація реалізована за допомогою Spring Security, а дані для авторизація витягуються із таблиці USER в базі даних DynamoDB. Після успішної аутентифікації і авторизації

генерується спеціальний код (token), котрий передається на додаток клієнт і містить в собі усю необхідну інформацію про користувача.

Підсистема завантаження відео представляє інтерфейс за допомогою якого користувач може завантажити своє відео для перевірки його оригінальності. Спочатку користувач виконує POST запит із власним відео в якості тіла запита, далі цей відеозапис завантажується на локальне сховище Docker контейнера, де працює Java додаток. Максимальний розмір відео – 200 MB, максимальна тривалість відео – 5 хвилин. Після цього відео перевіряється за допомогою бібліотеки JMF (Java Media Framework) – якщо відео відповідає усім вимогам і обмеженням, то йому видається унікальний ідентифікатор і він зберігається у сховище відео. Локальні файли після цих маніпуляцій видаляються.

Також існує другий сценарій роботи системи – навчання нейронної мережі під конкретного користувача. При обранні даної опції користувачеві видається завчасно підготовлений текст у форматі PDF, котрий він має роздрукувати, або відкрити на телефоні. Далі користувачу пропонується записати відеозапис з собою на веб камеру, в котрому він читає даний текст. Текст розрахований за розміром так, щоби його читання займало приблизно 5 хвилин. Важливо щоб користувач користувався саме роздрукованою версією тексту, або читав з телефона чи планшета, оскільки це потребує відволікання від камери і переводи погляду вниз, що покращує якість навчання нейронної мережі. Коли користувач зробить це відео, то він має його завантажити. Роздільна здатність відео має бути не менша за 720 p. Після завантаження відео відбувається найдовший процес – його обробка і навчання нейронної мережі з новим датасетом.

Для створення нового набору даних, відео користувача проходить обробку за допомогою бібліотеки Java Media Framework. Відеозапис розбивається на сегменти по 10 секунд, котрі накладаються один на одного. В такому разі, із запису довжиною в 5 хвилин можна отримати 290 сегментів по 10 секунд, які будуть архівовано, завантажено у сховище відео і передано до нейронної мережі

для її навчання. Даний процес є найдовшим у системі і може займати до 1 години часу в залежності від навантаження в цілому.

Підсистема роботи з нейронною мережею працює наступним чином. Дана система публікує і читає сервіс Amazon – SNS. Це брокер повідомлень, в котрому вже існує декілька створених топіків (topic) для повідомлень: learn-videos, videos-to-check, completed-videos. В залежності від задачі, котру треба вирішити, дана підсистема публікує повідомлення у топік learn-videos або у топік videos-to-check.

У перший топік потрапляє повідомлення з унікальним ідентифікатором архіву із згенерованим датасетом для навчання з його допомогою нейронної мережі. Система з нейронною мережею у свою чергу викачує цей датасет зі сховища відео і виконує навчання по ньому.

У другий топік під назвою videos-to-check потрапляють повідомлення з унікальним ідентифікатором відеозапису, котрий треба перевірити. Система з нейронною мережею викачує це відео і перевіяє його. Результат про оригінальність відео публікується у топік completed-videos, де вказано ідентифікатор відео і оригінальне воно чи підробка.

Підсистема роботи з нейронною мережею читає повідомлення у топіку completed-videos постійно, таким чином дізнаючись про перевірені відео на наявність підробки, а також оброблюються відео, котрі вже використано для навчання нейронної мережі. Такі відео видаляються зі сховища відео з двох причин. По-перше, зберігаючи особисті відео користувача система порушує їх конфіденційність і по-друге ці відео більше не потрібні, адже нейронна мережа вже пройшла тренування на них, тож можна їх позбавитись і зберегти ресурси сховища відео.

Підсистема роботи з базою даних створена за допомогою бібліотеки DynamoDB Spring Data. Ця бібліотека реалізує інтерфейс JPA, тому з її допомогою можна зручно працювати з сутностями бази даних. У розроблюваній системі є декілька сутностей – User і Video. Завдяки принципам NoSQL бази

даних, а в даному випадку DynamoDB, є змога не створювати складні структури бази даних, а використовувати принцип ключ-значення, і мати всього 1 таблицю – User, в якій є уся інформація про користувача і його завантажені відео. Ключем є унікальний ідентифікатор користувача, а значенням – JSON об’єкт з усією необхідною інформацією.

Підсистема прийому платежів основана на платіжній системі PayPal. Обрано саме цю систему оскільки планується вихід на міжнародний ринок при старті стартапу, а PayPal це міжнародний проєкт, котрий підтримується багатьма країнами. Реалізовано підсистему за допомогою бібліотеки PayPal Java SDK, яка дає змогу зручно працювати з API PayPal. Дана система призначена для отримання платежів по періодичним підпискам користувачів.

3.4 Тестування розробленої системи

Даний підрозділ описує результати роботи системи в цілому і нейронної мережі зокрема. Додаток завантажено на AWS EKS кластер, де він стабільно працює з виставленим ReplicaSet у значення 2. Тобто у кожен момент часу одночасно запущено як мінімум 2 додатки системи для кінцевого користувача і 1 репліка нейронної мережі.

Проведено тестування підсистем реєстрації і завантаження відео. Створено нового користувача з іменем автора дисертації (рис. 3.8), а також завантажити і перевірити два відеозаписи (рис. 3.9).

На рис. 3.8 зображений WEB інтерфейс Amazon DynamoDB, де видно, що в таблиці USER з’явився новий запис з ім’ям Andrew, прізвищем Barabash і ID 1. Це свідчить про коректну роботу системи реєстрації.

На рис. 3.9 також зображений WEB-інтерфейс сервісу Amazon, проте в цей раз – це Amazon S3. Як можна побачити створено окремий bucket з назвою possible-Deep Fake-videos, де знаходяться усі завантажені відеозаписи, які були коли-небудь завантажені користувачами.

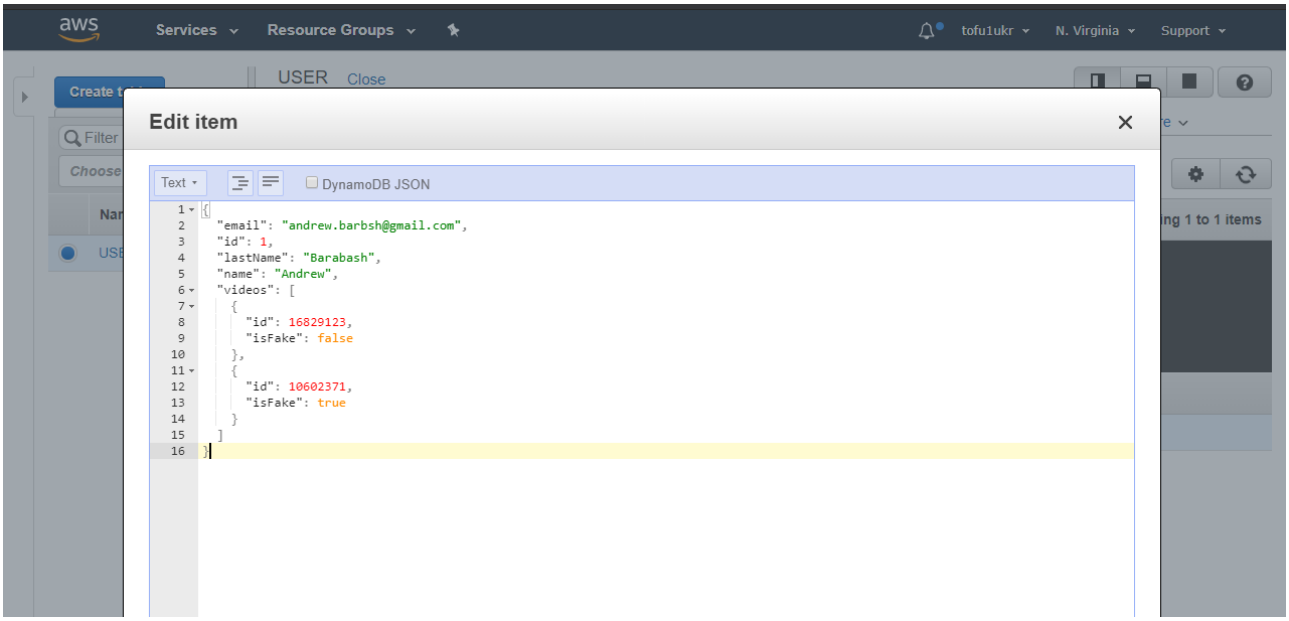


Рис. 3.8. Демонстрація успішного створення нового користувача

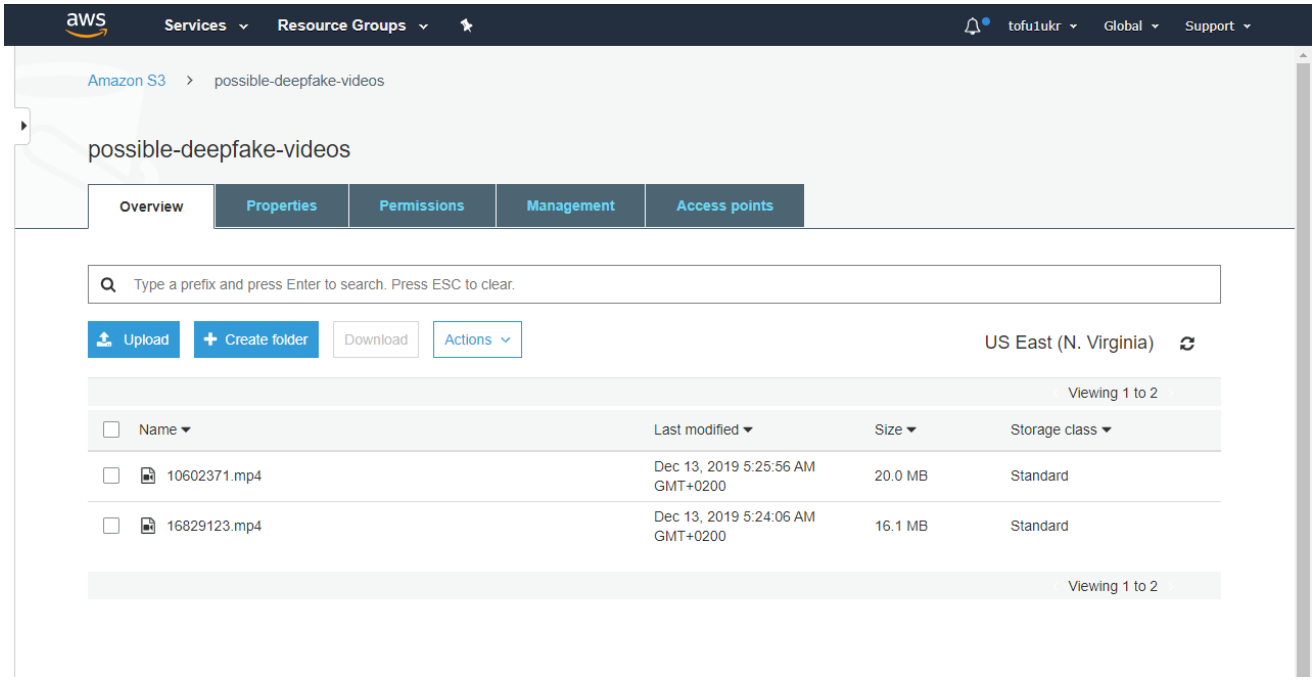


Рис. 3.9. Демонстрація успішного завантаження відео

На знімку екрана видно, що завантажено 2 відеозаписи, кожна з яких має назву унікального ідентифікатора, котрий видається підсистемою завантаження відео (id). Це свідчить про коректну роботу системи завантаження відео.

Тепер слід перейти до тестування самої нейронної мережі – головної системи розроблюваного додатка.

Працездатність кожної конкретної моделі ЦО перевіряється за допомогою комедійних імітаторів та глибоких підробок, специфічних для ЦО. Визначено точність тестування як площу під кривою (AUC) кривої робочої характеристики приймача (ROC) та істинно позитивної пропорції (ІПП) правильного розпізнавання оригіналу за істинно негативною пропорцією (ІНП) 1%, 5 % і 10%. Про ці точності повідомляється як для 10-секундних роликів, так і для сегментів повного відео. Відео сегмент класифікується на основі середньої оцінки SVM усіх 10-секундних кліпів, що перекриваються. Спочатку представлено детальний аналіз оригінальних і фальшивих відео Барака Обами, а потім аналіз інших ЦО.

У верхній половині табл. 2 показано точність класифікації відео Барака Обами за 190 характеристиками. Перші чотири рядки відповідають точності для 10-секундних кліпів, а наступні чотири рядки відповідають точності для сегментів повного відео. Середня AUC для 10-секундних кліпів та повних сегментів становить 0,93 та 0,98.

Найменша точність AUC у кліпу та сегменту для методу синхронізації губ - 0,83 та 0,93, ймовірно, тому що, порівняно з іншими підробками, ці підробки лише маніпулюють областю рота.

Як результат, багато міміки та рухів зберігаються в цих підробках. Як показано далі, однак, точність знаходження підробок зроблених методом синхронізації губ може бути підвищена за допомогою простої техніки обрізки.

Таблиця 3.1

Точність розпізнавання Deep Fake відео Барака Обами

	Випадкові люди	Комедійний імітатор	Обмін обличчям	Синхронізація губ	Майстер ляльок
190 ознак					
10-секундний кліп					
ІПП (1% ІНП)	0,62	0,56	0,61	0,30	0,40
ІПП (5% ІНП)	0,79	0,75	0,81	0,49	0,85
ІПП (10% ІНП)	0,87	0,84	0,87	0,60	0,96
AUC	0,95	0,94	0,95	0,83	0,97
Сегмент					
ІПП (1% ІНП)	0,78	0,97	0,96	0,70	0,93
ІПП (5% ІНП)	0,85	0,98	0,96	0,76	0,93
ІПП (10% ІНП)	0,99	0,98	0,97	0,88	1.00
AUC	0,98	0,99	0,99	0,93	1.00
29 ознак					
10-секундний кліп					
AUC	0,98	0,94	0,93	0,95	0,98
Сегмент					
AUC	1.00	0,98	0,96	0,99	1.00

Для вибору найкращих особливостей для класифікації, 190 моделей були ітеративно підготовлені маючи від 1 до 190 ознак. Зокрема, за першою ітерацією підготовлено 190 моделей із використанням лише однієї функції. Вибрано функцію, яка дала найкращу загальну точність тренувань. На другій ітерації 189 моделей підготовлено за двома ознаками, перша з яких була визначена за першою ітерацією. Далі обрано другу особливість, яка дала найкращу загальну точність тренувань для третьої ітерації. Весь цей процес повторився 190 разів. На рис. 3.10 показана точність тестування як функція кількості особливостей для перших 29 ітерацій цього процесу (точність тренування досягла максимуму при 29 ознаках).

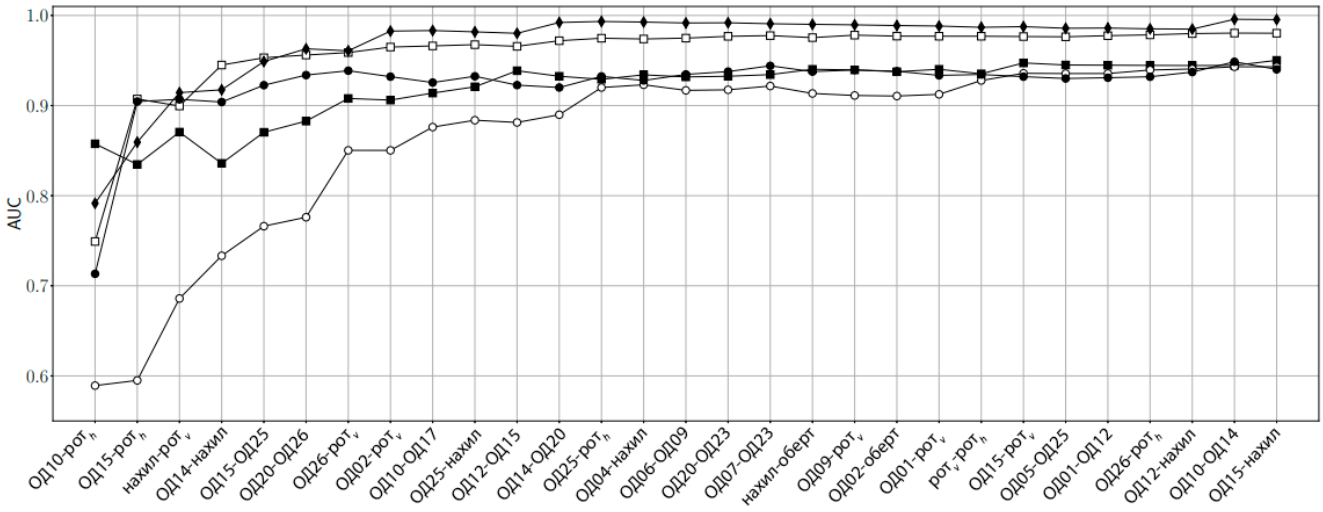


Рис. 3.10. Точність для різних методів штучних відео

На цьому рисунку відображена точність (AUC) для методів імітатора (чорний квадрат), випадкових людей (білий квадрат), синхронізації губ (чорний круг), обмін обличчям (білий круг), майстер ляльок (чорний ромб) для класифікатора тренованого на проміжку від 1 до 29 ознак, які перелічені на горизонтальній осі.

Маючи лише 13 функцій, AUC майже вийшла на плато з середнім показником 0,95. Також, точність починає повільно знижуватися після включення 30 функцій, але на даному рисунку це не показано. Першими п'ятьма

відмінними рисами є співвідношення між: (1) підйомом верхньої губи (ОД10) та 3-D горизонтальною відстанню між куточками рота ($рот_h$); (2) опусканням куточка губи (ОД15) та $рот_h$; (3) обертанням голови навколо осі X (кивання) та $рот_v$; (4) ямочкою (ОД14) і киванням; і (5) опусканням куточка губи (ОД15) і розведення губ (ОД25). Цікаво, що ці топ п’ять кореляцій мають принаймні один компонент, який відповідає ротовій порожнині. Припущено, що ці ознаки є найважливішими через характер підробок синхронізації, які лише змінюють область рота, а метод обміну обличчям, метод лялькового майстра та метод комедійних імітаторів просто не в змозі фіксувати тонкі рухи рота.

Як вже згадувалося раніше, багато криміналістичних методик виявляються невдалими внаслідок таких простих маніпуляцій, як рекомпресія, і тому було перевірено надійність розроблюваного підходу до цього виду захисту. Кожен оригінальний та фальшивий сегменти відео спочатку було збережено при якості квантування H.264 у 20. Кожен сегмент потім був рекомпресований з нижчою якістю 40. Далі приведені AUC для розрізнення 10-секундних кліпів Барака Обами від випадкових людей, комедійних імітаторів, обміну обличчям, синхронізації губ та лялькового майстра після цієї маніпуляції: 0,97, 0,93, 0,93, 0,92 та 0,96. Якість розпізнавання практично не змінюються у порівнянні з відео високої якості (табл. 3.1). Як і очікувалося, оскільки розроблювана технологія не покладається на артефакти на рівні пікселів, вона є надійною для простої маніпуляції відмивання.

Для того, щоб визначити стійкість до довжини кліпу, перенавчена чотири нові моделі, використовуючи кліпи довжиною 2, 5, 15 та 20 секунд. Середній показник AUC для всіх відео становить 0,80, 0,91, 0,97 та 0,98, порівняно з AUC 0,96 при тривалість кліпу 10 секунд. Як очікувалося, точність падає для коротших кліпів, але тривалість кліпу в основному не впливає при 10 та 20 секундах.

Стиль розмови та поведінка обличчя людини можуть змінюватися залежно від контексту, в якому людина розмовляє. Наприклад, поведінка обличчя під час

виступу підготовленої промови може істотно відрізнятися порівняно з відповіддю на напружене запитання під час інтерв'ю в прямому ефірі. У двох наступних експериментах перевірено надійність створеної моделі Обами проти різних контекстів, відмінних від тижневих розмов, що використовувались для тренувань.

У першому експерименті зібрано відео де, як і на щотижневих розмовах, Барак Обама спілкувався з камерою. Однак ці відео охоплювали різноманітні контексти, починаючи від оголошення про смерть Осама Бін Ладена до відеозапису президентських дебатів та рекламного відео. Загалом зібрано 1,5 години таких відеороликів, які дали 91 відео-сегмент тривалістю 1,3 години та 21152 кліпів тривалістю 10 секунд. Середня точність з точки зору AUC для розрізнення цих відеозаписів від комедійних імітаторів, випадкових людей, синхронізації губ та підробки методом майстра ляльок - 0,91 для 10-секундних кліпів і 0,98 для повних сегментів, попередня точність 0,96 і 0,99. Незважаючи на відмінності в контексті, здається, що розроблена модель досить добре працює з різними контекстами.

У другому експерименті зібрано інший набір відео Обами в ще більш істотно різних контекстах, починаючи від інтерв'ю, в якому він дивився на інтерв'юера, а не на камеру, до інтерв'ю в прямому ефірі, в якому він робив паузи значно більше під час своєї відповіді і часто дивився вниз. Зібрано загалом 4,1 години відео, які дали 140 відео-сегментів сумарною тривалістю 1,5 години та 19855 перекриваючих 10-секундних кліпів. Середній AUC значно знизився до 0,61 для 10-секундних кліпів і 0,66 для сегментів. У цьому випадку контекст відеороликів суттєво відрізнявся, так що вихідна розроблена модель не захоплювала необхідних функцій. Однак, перенавчивши модель на оригінальному наборі даних та цих відеороликах у стилі інтерв'ю, AUC збільшився до 0,82 та 0,87 за 10-секундні кліпи та сегменти. Незважаючи на вдосконалення, видно, що точність не така висока, як раніше, коли моделі треновано для ЦО та конкретних контекстів, тож в якості вдосконалення системи

слід розширювати поточні ознаки з більш стабільними та специфічними для ЦО характеристиками.

Проведено порівняння розробленої техніки з підходом CNN, який використовується в FaceForensics++, в якому кілька моделей пройшли навчання для виявлення трьох типів маніпуляцій обличчям, включаючи глибокі підробки обміну обличчям. Оцінено моделі з найбільш високими показниками, що були треновані використовуючи XceptionNet з обрізаними обличчями в якості вхідних даних. Продуктивність цих моделей протестована на реальній глибинній підробці відео Обами методом обміну обличчя, а також методом майстра ляльок, збережених у високих і низьких якостях (метод імітатора та випадкових людей не використовувався, так як вони не штучно синтезовані). Протестовано моделі, які доступні від авторів, без тонкого налаштування для нашого набору даних.

Вихідний кадр технології CNN використано для обчислення точності (AUC). Загальна точність виявлення кадрів підробок методом обміну обличчям, маріонеткового майстра та глибоких підробок із синхронізацією губ за якістю 20/40 становить 0,84/0,71, 0,53/0,76 та 0,50/0,50 порівняно з середнім AUC розробленої системи 0,96/0,94. Навіть незважаючи на те, що FaceForensics++ досить добре працює на глибоких підробках методом обміну обличчям, він не може упоратись із глибинними підробками методом синхронізації губ, котрі не використовувались під час навчального процесу.

Висновки до розділу

Отже, результатом роботи над даним розділом є аналіз існуючих алгоритмів розпізнавання обличчя, вибір такого алгоритму, опис створеного алгоритму детектування Deep Fake відеозаписів, створення нейронної мережі на основі цього алгоритму, створення системи для кінцевого користувача, а також тестування роботи нейронної мережі і системи в цілому, причому проведено

аналіз результатів роботи системи у порівнянні з доступними результатами конкурентів.

Для розпізнавання обличь обрано для використання згорткову нейрону мережу, тому що найкращі результати в області розпізнавання обличь показано саме нею.

Створено алгоритм для детектування Deep Fake відео. Він оснований на аналізі одиниць дії обличчя – унікальних рухів обличчя, котрі притаманні кожній людині. Такі рухи отримуються завдяки інструменту аналізу обличчя з відкритим кодом OpenFace2. За допомогою цього інструменту аналізуються 17 одиниць дії обличчя і порівнюються на схожість з оригіналом. На основі цього робиться припущення про оригінальність відеозапису. Інструментом створення нейронної мережі є мова програмування Python.

Розроблено систему для кінцевого користувача. Вона дозволяє користувачам отримати доступ до використання нейронної системи не відкриваючи деталі її реалізації. Розроблено підсистеми реєстрації, завантаження відео, роботи з базою даних, взаємодії з нейронною мережею, прийому платежів. Система розроблена за допомогою мови програмування Java і фреймворку Spring.

Результатом роботи нейронної мережі є розпізнавання штучно створених відео з точністю AUC 0,96/0,94 в залежності від якості наданих відеозаписів.

РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЄКТУ

Метою даного розділу є проведення маркетингового аналізу стартап-проєкту для визначення можливості його ринкового впровадження та можливих напрямів реалізації цього впровадження.

4.1 Опис ідеї проєкту

В табл. 4.1 викладено зміст ідеї, що пропонується , можливі напрямки застосування ідеї, основні вигоди, що може отримати користувач з ідеї та чим відрізняється ідея від існуючих аналогів.

Таблиця 4.1.

Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Система, яка визначає оригінальність відео	Використання системи для захисту від можливих наслідків від Deep Fake відео	Захист від можливих наслідків від штучних Deep Fake відеозаписів

Ця таблиця дає цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів. Основними вигодами користувача є підвищення якості обробки сигналів та розширення області застосування розробленої системи.

Далі проведемо аналіз потенційних техніко-економічних переваг даної ідеї порівняно із пропозиціями конкурентів. Визначимо чим вона відрізняється від існуючих аналогів та замінників. Результати наведено у табл. 4.2.

Таблиця 4.2.

Визначення сильних характеристик проєкту

№	Техніко-економічні характеристики ідеї	Мій проєкт	Конкуренти			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
			Berkley	USC ISI	SherlockAI			
1.	Відкритий доступ	+	-	-	-			+
2.	Перегляд попередніх завантажених відео	+	-	-	-			+
3.	Розпізнавання Deep Fake записів	+	+	+	+		+	

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

4.2 Технологічний аудит ідеї проєкту

В даному підрозділі проводиться аудит технології, за допомогою якої можна реалізувати ідею проєкту (технології створення товару). Технологічний аудит — операція об'єктивної оцінки потенціалу інновації як об'єкта комерціалізації. Через те, що комерціалізація технологій — тривалий і дорогий процес, те, перш ніж витрачати чималі тимчасові й фінансові ресурси, необхідно оцінити реальність продажу ідеї або винаходи або їхнє успішне перетворення в ринковий продукт. Таку оцінку можуть провести як самі автори, так і автори із залученням сторонніх експертів.

Визначення технологічної здійсненності ідеї проєкту передбачає аналіз складових, які наведено в табл. 4.3.

Таблиця 4.3

Технологічна здійсненність проєкту

Ідея проєкту	Технології її реалізації	Наявність технології	Доступність технологій
Захист будь-якого громадянина від шахрайства за допомогою Deep Fake відеозаписів	Згорткова нейронна мережа	Наявна	Доступна
	Сховище даних S3	Наявна	Доступна
	Docker	Наявна	Доступна
	Мова програмування Python і Java	Наявна	Доступна
	Платформа для розміщення системи в мережі Інтернет	Наявна	Доступна
	Система оркестрації Docker контейнерів	Наявна	Доступна
Обрана технологія реалізації ідеї проєкту: розробка нейронної мережі за допомогою Python і TensorFlow, розробка системи для кінцевого користувача за допомогою Java і Spring, обмін даними між системами за допомогою брокера повідомлень Amazon SNS, сховищем даних Amazon S3 і базою даних Amazon DynamoDB.			

4.3 Аналіз ринкових можливостей запуску стартап-проєкту

Цей підрозділ описує визначення ринкових можливостей, які можна використати під час ринкового впровадження проєкту, та ринкових загроз, які можуть перешкодити реалізації проєкту, дозволяє спланувати напрями розвитку проєкту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проєктів-конкурентів.

Таблиця 4.4

Попередня характеристика потенційного ринку проєкту

	Показники стану ринку	Характеристика
1.	Кількість головних гравців	2
2.	Динаміка ринку (якісна оцінка)	Зростає
3.	Наявність обмежень для входу (вказати характер обмежень)	Немає обмежень
4.	Специфічні вимоги до стандартизації та сертифікації	Сертифікація системи на предмет заявленої точності

Отже, за результатами аналізу таблиці можна зробити висновок, що ринок є привабливим для входження за попереднім оцінюванням.

У таблиці 4.5 визначено потенційні групи клієнтів та характеристики, після чого сформовано орієнтовний перелік вимог до товару для кожної з груп клієнтів. В даній таблиці визначено цільову аудиторію та її основні вимоги до товару даного виду, визначили основні характеристики системи обробки сигналів, які формують поведінку клієнтів відносно нашого приладу, та визначили основні вимоги до приладу.

Таблиця 4.5

Характеристика потенційних клієнтів стартап-проєкту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Розповсюдження Deep Fake відеоза-писів, потреба в захисті від них	Відомі особистості, політики, журналісти	Звичайні користувачі і правоохоронні служби	Доступність для кожного; Простота експлуатації; Ефективність; Висока точність; Доступна ціна

При застосуванні даної технології існують певні загрози. Для попередження таких ситуацій необхідно якісне обладнання, а також працювати з такими пристроями повинні висококваліфіковані фахівці. Також, повинно своєчасне технічне оновлення даного продукту. Після визначення потенційних груп клієнтів проводимо аналіз ринкового середовища. Складаємо таблиці факторів, що сприяють ринковому впровадженню проєкту, та факторів, що йому перешкоджають (табл. 4.6 і табл. 4.7).

Таблиця 4.6

Фактори можливостей

Фактор	Зміст можливості	Можлива реакція компанії
Конкуренція	Ліцензування	Отримання ліцензії для захисту від плагіату і збільшення довіри користувачів
Попит	Збільшення попиту на даний вид товару	Збільшення продаж та впровадження знижок

Таблиця 4.7

Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1.	Конкуренція	Вихід на ринок конкурента	Знизити ціну на платну підписку; Запропонувати безкоштовні перевірки кожного тижня
2.	Технічний	Збільшення часу при необхідній точності обробки даних.	Покращення алгоритмів перевірки відео
3.	Постачання	Невчасне постачання компонентів	Пошук нових постачальників

В таблицях 4.6 та 4.7 наведено основні фактори що загрожують та сприяють ринковій впровадженні даного проєкту.

В табл. 4.8 проводиться аналіз пропозиції, тобто визначаються загальні риси конкуренції на ринку. Найважливішими ознаками, за якими виділяють різні моделі ринку, є:

- кількість фірм-продавців на ринку;
- тип продукту, що пропонується для продажу;
- можливості контролю за цінами з боку продавців;
- умови вступу в галузь додаткових виробників та виходу з неї;
- метод конкуренції, який переважає на цьому ринку.

З'ясування загальних ознак конкурентного ринку та особливостей функціонування на ньому фірми і формування її доходів дає досить підстав для роз-робки моделі вибору фірмою обсягів виробництва, які забезпечують їй

максимальний прибуток. Така модель має свою специфіку для короткотермінового та довго-термінового періодів

Таблиця 4.8

Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції: олігополія	Існування невеликої кількості компаній, що працюють в даній сфері	Якісна продукція, правильна цінова політика, висока якість обслуговування
2. За рівнем конкурентної боротьби: міжнародна	Представники в різних країнах	Вихід на міжнародний ринок
3. За галузевою ознакою: одногалузева	Підприємства працюють в межах однієї галузі	Пропозиція товару за більш низькою ціною
4. Конкуренція за видами товарів: товарно-видова	Пропонують товари одного виду	Реклама, простота в користуванні
5. За характером конкурентних переваг: якісна	Вартість залежить від ціни на матеріали та комплектуючі елементи	Вибір оптимального варіанту ціна/якість комплектуючих елементів

Після ступеневого аналізу конкуренції в табл. 4.9 проводиться більш детальний аналіз умов конкуренції в галузі за моделлю 5 сил М. Портера.

Таблиця 4.9

Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Потенційні конкуренти в галузі	Постачальники	Клієнти	Товари-замінники
	Економія на масштабуванні	Значення розміру максимального масштабування	Контроль якості	Лояльність споживачів
Висновки	Є можливість виходу на ринок за рахунок гнучкого масштабування	Постачальники не впливають на умови роботи на ринку	Клієнти вимагають якісної та простої в керуванні продукції	Більш відомі компанії захоплюють ринок

За результатами аналізу даних таблиць можна зробити висновок, що в даний момент з огляду на конкурентну ситуацію щоб вийти на ринок та бути конкурентоспроможним проєкт повинен не поступатись в точності виявлення підробок, але в той же час повинен пропонуватися за дещо нижчою ціною ніж в конкурентів. В табл. 4.10 наведено фактори конкурентоспроможності, які засновані на аналізі конкуренції, який проведений в табл. 4.9, а також із урахуванням характеристик ідеї проєкту (табл. 4.2), вимог споживачів до товару (табл. 4.5) та факторів маркетингового середовища (табл. 4.6 і табл. 4.7).

Конкурентоспроможність це здатність певного об'єкта або суб'єкта перевершити конкурентів у заданих умовах. Визначимо та обґрунтуємо перелік факторів конкурентоспроможності у табл. 4.10.

Таблиця 4.10

Обґрунтування факторів конкурентоспроможності

Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проєктів значущим)
Наявність сертифікатів та патенту	Наявність таких документів не дає можливості використання нашої технології конкурентами
Простота в експлуатації	Можливість експлуатації без спеціального навчання для клієнтів
Якість продукту	Висока якість продукту дозволить проєкту швидко закріпитись на ринку
Ціна	Визначення стану ціни на ринку та встановлення її дещо нижчою, ніж в конкурентів
Сервісне обслуговування	Якісне обслуговування приваблює клієнтів

Таким чином, в табл. 4.10 визначено основні фактори конкурентоспроможності, за допомогою яких далі проведено аналіз сильних та слабких сторін стартап-проєкту.

На основі цього проведено аналіз сильних та слабких сторін. Цей порівняльний аналіз дозволяє зрозуміти сильні і слабкі сторони конкурентів, котрі вже представлені на ринку, що дає змогу підготувати стартап-проєкт до конкурентоспроможного виходу на ринок. Даний аналіз показано в табл. 4.11.

Таблиця 4.11

Порівняльний аналіз сильних та слабких сторін

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів						
			-3	-2	-1	0	1	2	3
1	Простота експлуатації	17	+						
2.	Ціна	15					+		
3.	Сервісне обслуговуваання	16		+					
4.	Наявність сертифікатів та патенту	18							+
5.	Якість продукту	17		+					

З таблиць 4.10 та 4.11 можна побачити, що фактори конкурентоспроможності є досить хорошими. Основною перевагою та головним досягненням є виконання контролю витрат енергії за рахунок зменшення дискретизації, висока якість продукту та сервісне обслуговування протягом всього терміну його використання споживачем. Фінальним етапом ринкового аналізу можливостей впровадження проєкту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities), що наведені в табл. 4.12, на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін , що наведені в табл. 4.11.

Традиційний метод SWOT – аналізу дозволяє провести детальне дослідження зовнішнього й внутрішнього середовища. Результатом раціонального SWOT-аналізу, спрямованого на формування узагальненого інформаційного потенціалу, повинні з'явитися ефективні рішення, що

стосуються відповідної реакції (впливу) суб'єкта (слабкої, середньої й сильної) відповідно до сигналу (слабкому, середньому або сильному) зовнішнього середовища.

Таблиця 4.12

SWOT аналіз стартап-проєкту

Сильні сторони:	Слабкі сторони:
Зменшення витрат енергії за рахунок зменшення дискретизації. Собівартість продукції нижча, ніж в конкурентів, якісне сервісне обслуговування	Залежність ціни продукту і довіри клієнтів від точності і непередбачуваних ситуацій
Можливості:	Загрози:
Вихід за закріплення на ринку, вихід на міжнародний ринок, отримання держзамовлення, збільшення попиту, ліцензійні договори	Інфляція, поява нових конкурентів на ринку, зростання цін, нестабільний хостинг, вихід з ладу системи чи перебої і точності виміру

Таблиця 4.13

Альтернативи ринкового впровадження стартап-проєкту

№	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Отримання держзамовлення	Висока	2 місяці
2.	Пошук інвестицій	Висока	24 місяці

Таким чином, розглянувши можливості ринкового впровадження стартап-проєкту можна зробити висновок, що основною альтернативою є отримання держзамовлення на прилад, оскільки ймовірність отримання ресурсів висока, а терміни реалізації менші.

4.4 Розроблення ринкової стратегії проєкту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 4.14).

Таблиця 4.14

Вибір цільових груп потенційних споживачів

Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
Приватні та державні автомобільні компанії	Продукт потрібний	Високий	Помірна	Необхідність мати ліцензію

Які цільові групи обрано: Провівши аналіз цільових груп споживачів було прийнято рішення співпрацювати як з приватними і державними компаніями

За результатами аналізу потенційних груп споживачів обрано цільові групи, для яких будемо пропонувати свій продукт та визначили стратегію охоплення ринку: стратегію масового маркетингу, із всім ринком, пропонуючи стандартизовану програму. Для роботи в обраних сегментах ринку сформуємо базову стратегію розвитку (табл. 4.15).

Теорія і практика підприємницької діяльності виділяють три види базових економічних стратегій - це виживання, стабілізація та розвиток. Кожна з цих стратегій визначається рівнем досягнутої (запланованої) рентабельності та життєвим циклом товарів (послуг).

Таблиця 4.15

Визначення базової стратегії розвитку

Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Удосконалення даного проєкту	Диференційований маркетинг	Якість, точність виміру, ціна	Диференціації

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 4.16) та стратегії позиціонування (табл. 4.17).

Таблиця 4.16

Визначення базової стратегії конкурентної поведінки

Чи є проєкт «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Так	Шукати нових	Ні, в цьому немає необхідності	Стратегія виклику лідеру

Таблиця 4.17

Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап-проєкту	Вибір асоціацій, які мають сформувати комплексну позицію власного проєкту
Ціна, Простота використан ня, Точність, Надійність	На основі специфічних характеристик	Ціна, точність, швидкість та плавність	Позиціювання «Ціна-якість»

На основі вимог споживачів з обраного сегменту до постачальника і продукту, а також в залежності від стратегії розвитку та стратегії конкурентної поведінки розробляємо стратегію позиціювання, яка визначається у формування ринкової позиції, за яким споживачі мають ідентифікувати проєкт.

4.5 Розроблення маркетингової програми стартап-проєкту

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у табл. 4.18 сформуємо результати попереднього аналізу конкурентоспроможності товару за основними показниками:

- потреба;
- висока точність;
- низька ціна.

Надалі розробляється трирівнева маркетингова модель товару: уточнимо ідеї продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл. 4.19).

Таблиця 4.18

Визначення ключових переваг потенційного товару

Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
Висока точність	Похибка результатів становить тільки 1%	Використання азимуту для розрахунку ймовірності місцезнаходження
Низька ціна	Пропонування товару за дещо нижчою ціною ніж в конкурентів	Збільшення кількості клієнтів за рахунок ціни та якості продукції

Таблиця 4.19

Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові	
I. Товар за задумом	Система, яка реалізує детектування підроблених відеозаписів	
II. Товар у реальному виконанні	Властивості/характеристики	Значення
	1.Можливість контролювати затрати енергії	+
	2. Похибка результатів	1%
	3. Можливість передачі даних в режимі «online»	+
	Марка: «Покращена система обробки сигналів»	
	Якість: ГОСТ 14782-86 контроль неруйнівний	
	Пакування	
III. Товар із підкріпленням	До продажу включається	
	Після продажу включається	

За рахунок цього потенційний товар буде захищено від копіювання: За рахунок оформлення патенту (отримання сертифікату про інтелектуальну власність) та надання кожному проданому приладу індивідуального ліцензійного коду який необхідний для авторизації та підключення до системи передачі даних.

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на платну підписку (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (табл. 4.20). Аналіз проводиться експертним методом.

Таблиця 4.20

Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
Товари-замінники відсутні	Індивідуальні для кожного, в районі 50000 грн за рік	Від 100000 грн	1000-3000 грн за місяць

В даній таблиці проаналізовано ринкові ціни на товари аналоги та замінники, а також середній рівень доходів споживачів. За отриманими даними встановлена верхня та нижня межа ціни на місячну підписку.

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 4.21). Проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту). Вибір та обґрунтування оптимальної глибини каналу збуту; вибір та обґрунтування виду посередників.

Таблиця 4.21

Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Продаж в роздріб та спецзамовлення	Безпосередній продаж товару клієнту.	Висока	Спец та держзамовлення

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Таблиця 4.22

Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Спостереження за новинками на ринку, Замовлення товарів онлайн	Публікації Інтернет виставки	Ефективність, ціна, простота використання, точність	Донести переваги даної продукції	Контролюй витрати енергії.

Результатом є ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого буде впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

Висновки до розділу

Аналіз стартап-проєкту показав можливість ринкової комерціалізації продукту. Це обумовлено технологічною новизною проєкту і точно підібраною цільовою аудиторією не дивлячись на наявну конкуренцію певного рівня. Продукт здатен подолати бар'єри входження в ринок, такі як необхідність отримання ліцензування, необхідність отримати патент та інтелектуальну власність на алгоритм, а також отримати держзамовлення. Продукт одразу виходить на міжнародний ринок, тому економіка і закупівельна поведінка окремих країн не важлива для успішного просування проєкту.

Треба враховувати, що на ринку вже присутні аналоги розроблюваного продукту. Проте даний проєкт являється конкурентоспроможним через новизну і покращену точність алгоритму порівняно з іншими, а також можливістю працювати на масовий ринок. Звісно, держ- і спецзамовлення принесуть набагато більше прибутків, аніж ринок масового збуту, проте якщо отримати відомість серед звичайних користувачів, то буде легше отримати ті самі замовлення по індивідуальним цінам. Для високопоставлених замовників діють спеціальні умови, за якими індивідуально створюється датасет для конкретної особи, проводиться навчання нейронної мережі і виділяється окремий сервер. Таким чином перевірка відео на оригінальність відбувається одразу за запитом без затримки. Тому для успішного виходу стартап-проєкту на ринок потрібні:

- Гнучка інфраструктура для доступу до продукту з будь-якої точки світу;
- Реклама для отримання перших клієнтів;
- Результати реальних клієнтів і добрі відгуки;
- Отримання перших держ- і спецзамовлень.

Після досягнення останнього пункту стартап-проєкт перестає бути стартапом і стає справжнім бізнесом, що генерує прибуток. На даному етапі варто зосередитись на корпоративному ринку, розумному маркетингу, а також на державних замовленнях.

ВИСНОВКИ

В ході виконання магістерської дисертації розглянуто питання захисту громадян від штучно зроблених відео. Проблема є актуальною, оскільки на сьогоднішній день розвиток нейронних мереж є настільки швидким, що з'явились системи, що можуть нашкодити людству. Однією з таких технологій є Deep Fake (глибинні фейки) – відеозаписи на яких обличчя людини заміняється на обличчя іншої людини, причому усі емоції, вирази обличчя і рухи не зникають, а підробку майже не помітно. Розроблена в дисертації система детектування Deep Fake відеозаписів націлена вирішити дану проблему.

Проаналізовано існуючі рішення виявлення підробних відеозаписів. Здебільшого вони базуються на недоліках, які були на штучних відеозаписах тоді, коли дані системи розроблювались. Наприклад, система розроблена командою з каліфорнійського інституту USC ISI заснована на стеженні за нехарактерними для реальної людини рухами обличчя. Такі системи не будуть працювати сьогодні, адже якість штучних відео швидко зростає.

Для розпізнавання обличчя із зображення в дисертації використано згорткові нейронні мережі. Цей метод демонструє найкращі результати по розпізнаванню обличчя на сьогоднішній день, що підтверджено результатами відомого конкурсу із розпізнавання обличь ImageNet. Для вимірювання і відстеження обличчя використано інструмент OpenFace2, за допомогою якого виділено 17 одиниць дії обличчя, що далі аналізуються за допомогою кореляції Пірсона. Нейронна мережа створена за допомогою мови програмування Python.

Для перевірки працездатності нейронної мережі вона була навчена за допомогою власноруч створеного датасету. В нього входять відео цільових осіб, які були завантажені з YouTube і розбиті на сегменти по 10 секунд.

Створено систему для кінцевого користувача, завдяки якій можна отримати доступ до використання нейронної мережі без розкриття її реалізації. Розроблено підсистеми реєстрації, завантаження відео, роботи з базою даних,

взаємодії з нейронною мережею і прийому платежів. Система розроблена за допомогою мови програмування Java і фреймворку Spring.

Інфраструктура розробленої системи разом із нейронною мережею працює завдяки сервісам від компанії Amazon. Для взаємодії між системою і нейронною мережею для кінцевого користувача використано Amazon SNS, в якості бази даних обрано Amazon DynamoDB, сховищем відео виступає Amazon S3, а працює вся система на Amazon EKS кластері, котрий керується за допомогою Kubernetes.

Розроблена система має кращі результати із розпізнавання Deep Fake відеозаписів за існуючих на ринку конкурентів. Лідером на сьогодні є система від компанії CNN під назвою FaceForensics++, яка має в середньому точність AUC 0,84/0,71 в залежності від якості відео. Розроблена в дисертації система має точність AUC 0,96/0,94, що підтверджено при аналізі однакових відеозаписів.

Проведено маркетинговий аналіз стартап-проєкту. Здійснено опис ідеї проєкту, технологічний аудит ідеї проєкту, аналіз ринкових можливостей запуску стартап-проєкту. Розроблено ринкову стратегію проєкту та маркетингову програму стартап-проєкту.

ПЕРЕЛІК ПОСИЛАНЬ

1. Shymanskyi M.O., Kornaga Y.I., Barabash A.O. System for Text Localization on Images Using Convolutional Neural Network. East European Scientific Journal, #4 (32), 2018. Part 1. Warsaw, Poland. P. 51 – 55.
2. Михеєв Ю.І., Барабаш А.О. Автоматизація процесу розробки спеціального контенту. Науковий журнал «Наукоємні технології». К.: НАУ, 2018. Том. 38. № 2. С. 169 – 176.
3. Вітюк А.Є., Корнага Я.І., Барабаш А.О. Захоплення невідомих об’єктів мобільним роботом із використанням візуальної інформації. Науковий журнал «Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки». Том 29 (68). № 1. 2018. Частина 1. С. 93 – 98.
4. Laptev A., Kliukovskyi D., Barabash A., Zidan A. Analysis of Existing Signal Detection Methods, Development of a Technique for Calculating the Probability of Secret Information Capture. International Journal of Science and Engineering Investigations (IJSEI). Denmark, 2019. Volume 8, Issue 92. P 99 – 103. http://www.ijsei.com/archive-89219.htm
5. Dovzhenko N.M., Salanda I.P., Barabash A.O., Koval M.O. Investigation of the method of transmission of information in wireless sensor networks between intelligent sensor nodes. Science and Education a New Dimension. Natural and Technical Sciences, 2019, No VII(23), Issue 193. P. 39 – 41. https://seanewdim.com/published-issues.html
6. Лаптєв О.А., Половінкін І.М, Клюковський Д.В., Барабаш А.О. Модель пошуку засобів негласного отримання інформації на основі диференціальних перетворень. Sciences of Europe. Praha, Czech Republic (ISSN 3162-2364). 2019. Vol. 1. No 43. P.59-62.
7. Корнага Я., Шиманський М., Барабаш А. Локалізація тексту на зображеннях за допомогою згорткових нейронних мереж. Міжнародна

науково-технічна конференція “Security, Fault Tolerance, Intelligence” ICSFTI 2018. м. Київ, 10-12 травня 2018 р. К.: НТУУ «Київський політехнічний інститут імені Ігоря Сікорського». С. 289 – 293. http://comsys.kpi.ua

8. Барабаш А. О., Мусієнко А. П. Методика діагностування нестійких відмов та збоїв бездротових сенсорних мережах. Матеріали науково-технічної конференції «Інноваційні аерокосмічні технології в екологічному моніторингу», м. Київ, 24-25 квітня 2018 р. К.: ДЕА, 2018. С.11.
9. Барабаш А.О., Коваль М.О. Використанням принципу справедливості під час обслуговування сенсорних мереж Десята міжнародна науково-практична конференція «Інтегровані інтелектуальні робототехнічні комплекси (ІІРТК-2018)». Збірка тез, м. Київ, 22-23 травня 2018 року. К.: НАУ, 2018. С. 28 – 30.
10. Лаптєв О.А., Клюковський Д.В., Барабаш А.О., Методика розрахунку ймовірності негласного отримання інформації на основі існуючих методів виявлення сигналів. Тези доповіді 52 міжнародної конференції «Розвиток науки в XXI столітті», м. Харків, 14 вересня 2019 р. Харків: НТУ «ХПІ», 2019. С.62 – 74.
11. G.J. Edwards, T.F. Cootes, C.J. Taylor. Face Recognition Using Active Appearance Models. Computer Vision - ECCV'98, 5th European Conference on Computer Vision, Freiburg, Germany, June 2-6, 1998, Proceedings, Volume II.
12. P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: 19, Issue: 7, Jul 1997, pp. 711 – 720.
13. Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, Lior Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. 2014 IEEE Conference on Computer Vision and Pattern Recognition (ISBN 978-1-4799-5118-5).

ДОДАТОК А

Графічні матеріали

Архітектура системи

Мережа інтернет

Front End

- React SPA
- Javascript
- CSS
- HTML

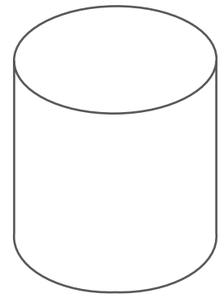
Back End

- Java Media Framework
- AWS SDK
- JPA
- Spring Framework
- Java

Нейронна мережа

- AWS SDK
- Tensorflow
- Python

База даних



Amazon DynamoDB

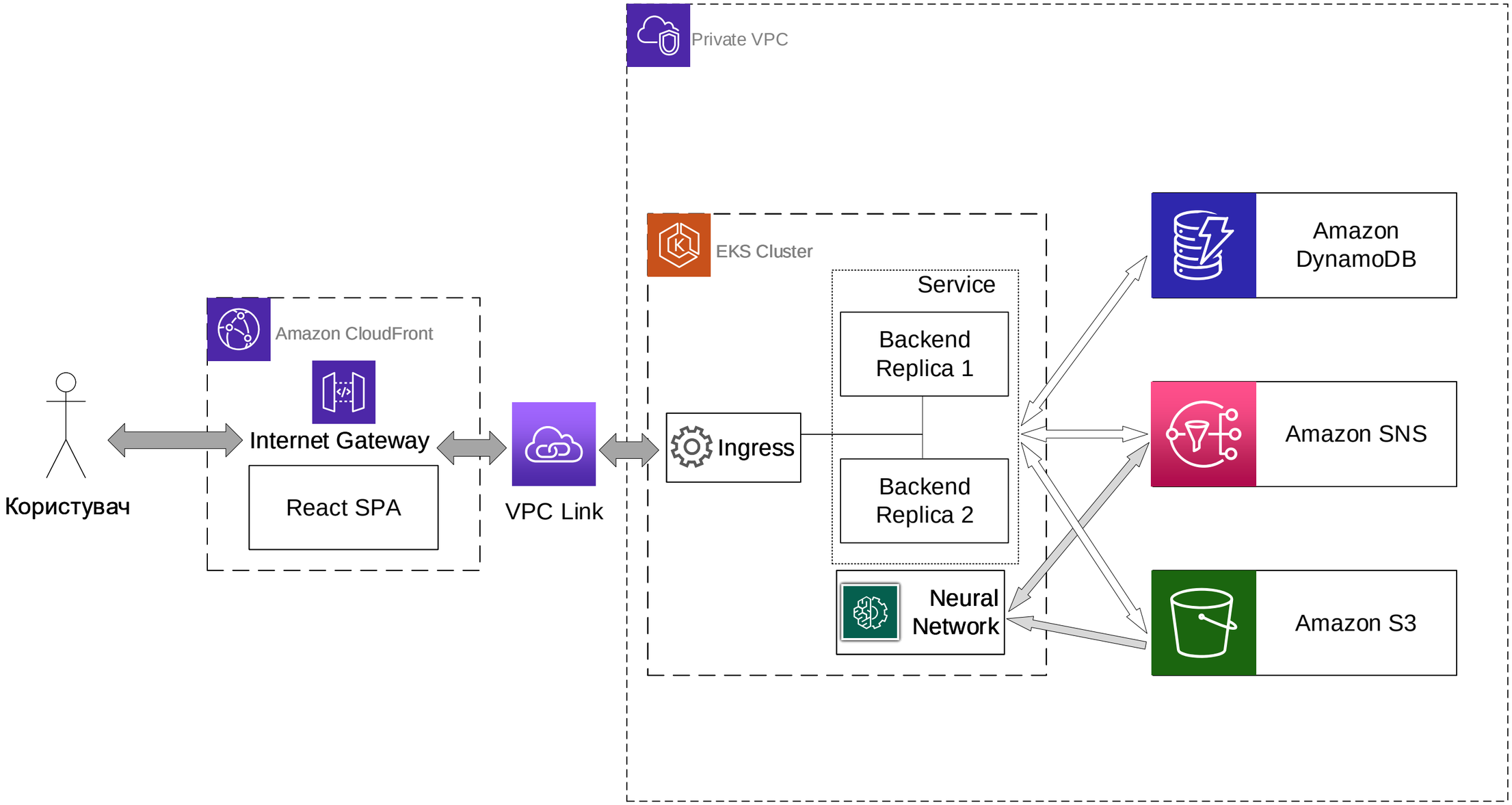
Сховище відео

Amazon DynamoDB

Демонстраційний плакат № 1
до дипломної роботи на тему
„Система детектування DeepFake відеозаписів на основі нейронної мережі”

Розробив: Барабаш А. О.
Прийняв: Корнага Я. І.

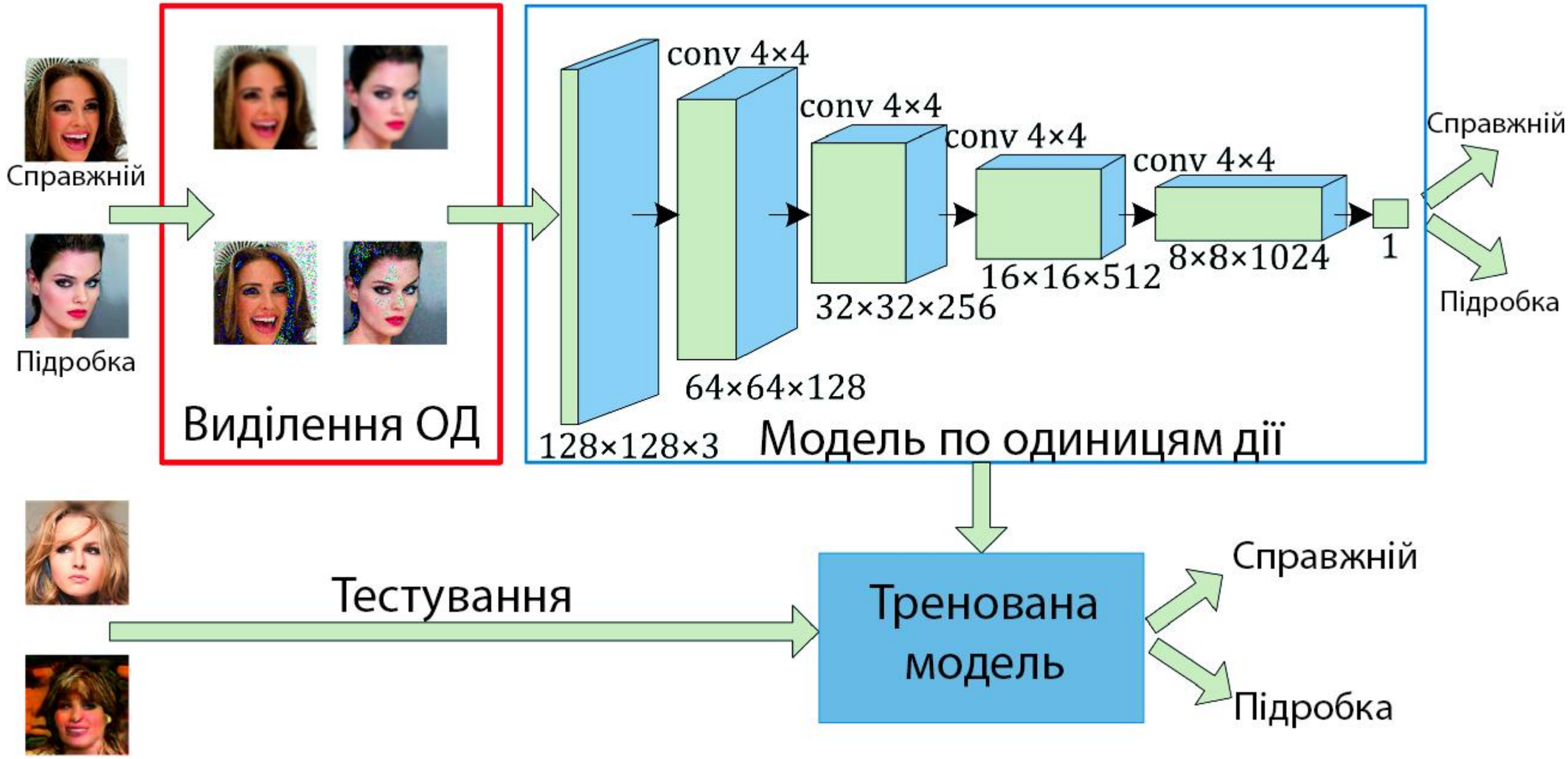
Інфраструктура системи



Демонстраційний плакат № 2
до дипломної роботи на тему
„Система детектування DeepFake відеозаписів на основі нейронної мережі”

Розробив: Барабаш А. О.
Прийняв: Корнага Я. І.

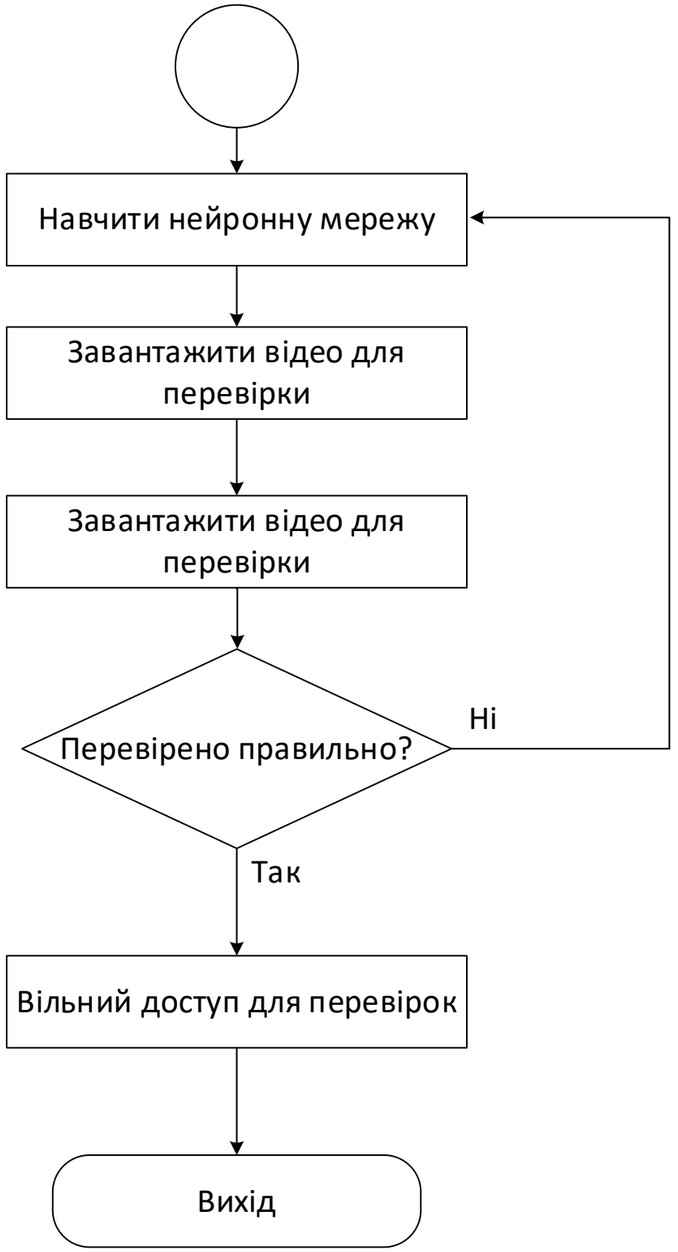
Принцип роботи ЗНМ



Демонстраційний плакат № 3
до дипломної роботи на тему
„Система детектування DeepFake відеозаписів на основі нейронної мережі”

Розробив: Барабаш А. О.
Прийняв: Корнага Я. І.

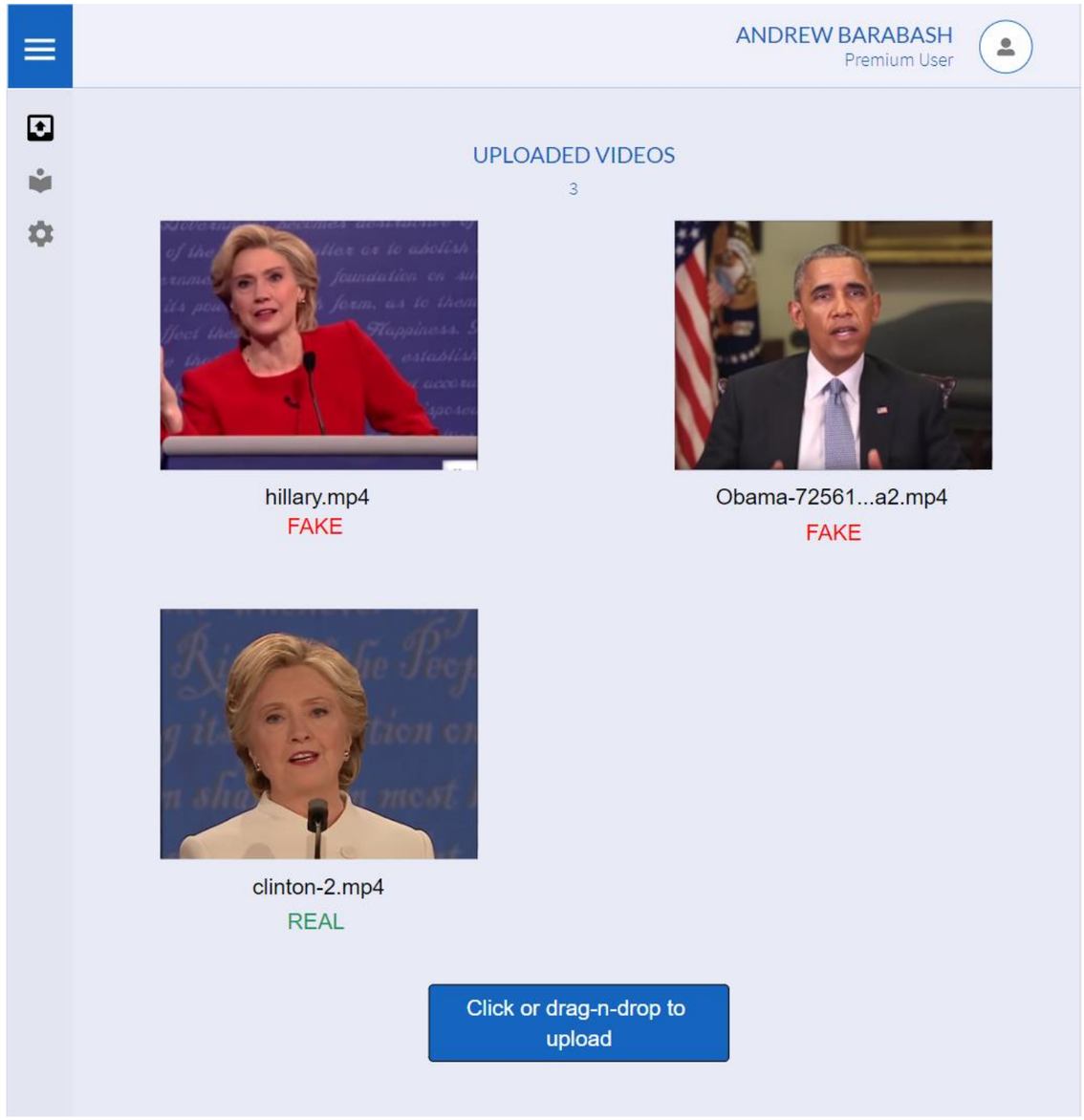
Алгоритм роботи користувача

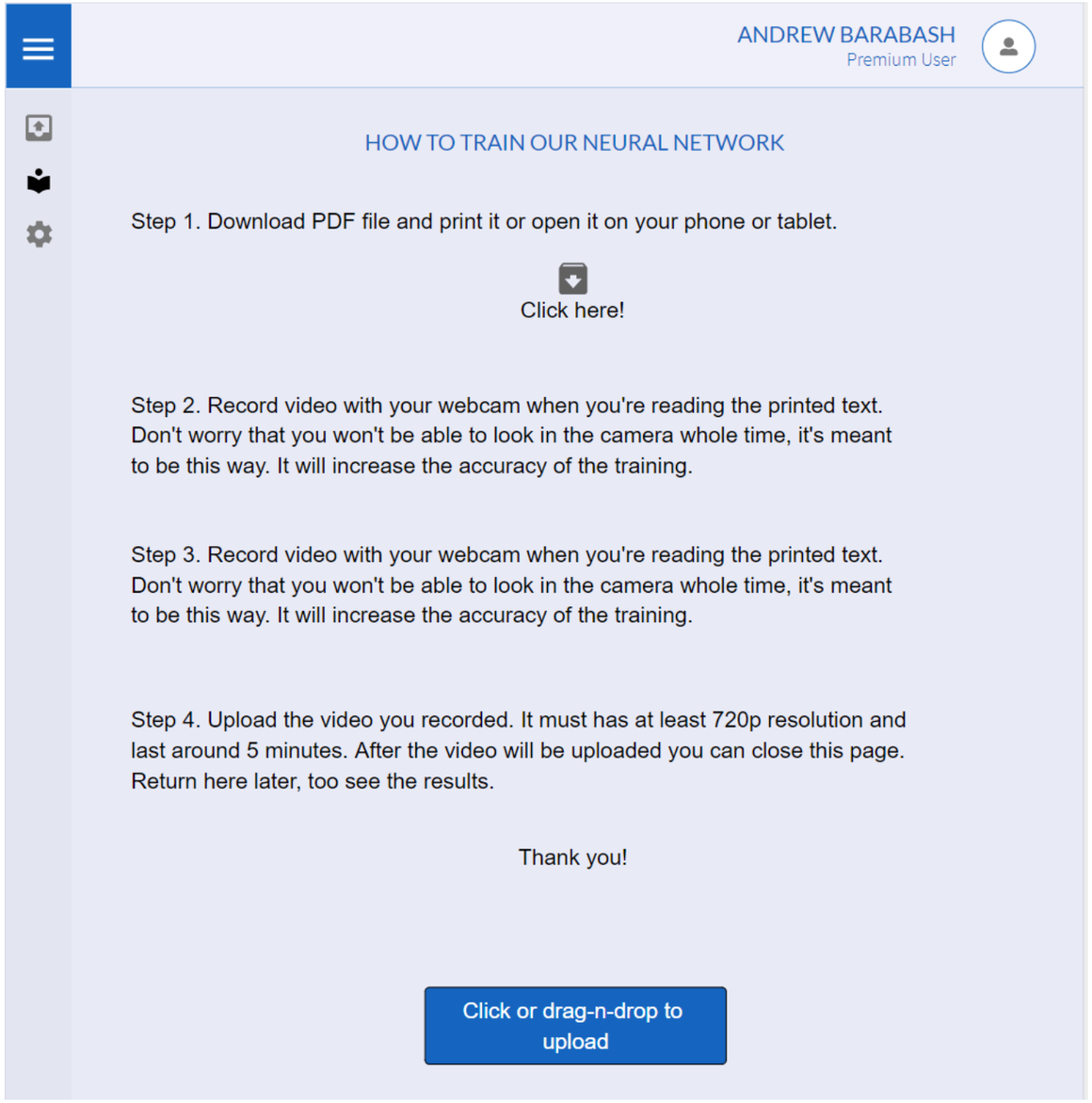


Демонстраційний плакат № 4
до дипломної роботи на тему
„Система детектування DeepFake відеозаписів на основі нейронної мережі”

Розробив: Барабаш А. О.
Прийняв: Корнага Я. І.

Інтерфейс системи





Демонстраційний плакат № 5
до дипломної роботи на тему
„Система детектування DeepFake відеозаписів на основі нейронної мережі”

Розробив: Барабаш А. О.
Прийняв: Корнага Я. І.

Результат роботи нейронної мережі

Точність розпізнавання Deep Fake відео Барака Обами

	Випадкові люди	Комедійний імітатор	Обмін обличчям	Синхронізація губ	Майстер ляльок
190 ознак					
10-секундний кліп					
ІПП (1% ІНП)	0,62	0,56	0,61	0,30	0,40
ІПП (5% ІНП)	0,79	0,75	0,81	0,49	0,85
ІПП (10% ІНП)	0,87	0,84	0,87	0,60	0,96
AUC	0,95	0,94	0,95	0,83	0,97
Сегмент					
ІПП (1% ІНП)	0,78	0,97	0,96	0,70	0,93
ІПП (5% ІНП)	0,85	0,98	0,96	0,76	0,93
ІПП (10% ІНП)	0,99	0,98	0,97	0,88	1.00
AUC	0,98	0,99	0,99	0,93	1.00
29 ознак					
10-секундний кліп					
AUC	0,98	0,94	0,93	0,95	0,98
Сегмент					
AUC	1.00	0,98	0,96	0,99	1.00

1 – метод «Випадкові люди»;

2 – метод «Комедійний імітатор»;

3 – метод «Обмін обличчями»;

4 – метод «Синхронізація губ»

5 – метод «Майстер ляльок»

Демонстраційний плакат № 6
до дипломної роботи на тему
„Система детектування DeepFake відеозаписів на основі нейронної мережі”

Розробив: Барабаш А. О.
Прийняв: Корнага Я. І.

ДОДАТОК Б

Результат перевірки на співпадіння